برمجة الحاسبات الشخصية
بلغة البيسك
M S X

رياض عزيز هرزه

برمجة الحاسبات الشخصية
بلغة البيسك
M S X

رياض عزيز مرزه

● بكالوريوس فيزياء من جامعة بغداد 1976.

● يعمل في مجال الليزر والبصريات، وله العديد من البحوث والمحاضرات في هذا المجال.

● فاز كتابه «مبادىء واستخدامات الليزر» بالجائزة الثالثة في مسابقة ديوان رئاسة الجمهورية الخاصة بتأليف الكتب العلمية. وحاز على تكريم السيد رئيس الجمهورية.

الإهْـداء

إلىٰ ولدي الحبيب علي . .

إلىٰ ابنتي الحبيبة فرح . .

رياض عزيز

المحتويات

الموضوع	الصفحة
المقدمة	9
الفصل الأول: تعاريف عامة	11
الفصل الثاني: لوحة المفاتيح	13
الفصل الثالث: تشغيل الحاسب	17
الفصل الرابع: الطور المباشر	25
الفصل الخامس: تصحيح الأخطاء	29
الفصل السادس: رسائل الخطأ	33
الفصل السابع: مبادىء ا لبرمجة	39
الفصل الثامن: فن التشكيل	77
الفصل التاسع: الأشكال المتحركة	105
الفصل العاشر: الصوت والموسيقى	153
الملاحق:	
ملحق (1) البيسك العربي (خاص بالحاسب «صخر»)	163
ملحق (2) إيعازات البيسك	171
ملحق (3) الإيعازات الخاصة بالحاسب «NEC»	201
ملحق (4) جدول بالرموز الإنكليزية المستخدمة في حاسبات «MSX»	205
ملحق (5) جدول بالرموز الغربية والانكليزية المستخدمة في «صخر بيسك»	207
ملحق (6) جدول بالرموز العربية والانكليزية المستخدمة في «صخر MSX»	209
ملحق (7) جدول بالرموز الانكليزية المستخدمة في الحاسب «NEC»	211
ملحق (8) نظام الأرقام	213

الموضوع	الصفحة

جدول المصطلحات المستخدمة في الكتاب 215
المصادر 221

بسم الله الرحمن الرحيم

المقدّمة

أهلاً بك عزيزي القارىء في عالم الحاسبات وأساليب البرمجة. إن الحاسب الإلكتروني قد خدم الناس منذ اليوم الأول لاختراعه، ولكنه كان، ولسوء الحظ، مكلفاً جداً ويحتاج إلى رصيد عالٍ لبنائه، هذا بالإضافة لكونه يستهلك قدرة عالية جعلت عملية استخدامه أمراً غير عملي، ولذلك ذهب المصممون والمهتمون لتطوير الحاسب الآلي ليكون بشكله البيتي البسيط الذي تراه بين يديك.

إن التعامل مع الحاسب يحتاج إلى دراية كافية من المستخدم في علوم الرياضيات، وفكرة عامة عن العلوم الأخرى، ولذلك بذلت جهود كبيرة لإنتاج حاسبات صغيرة ورخيصة وسهلة الاستعمال، وذات استخدام عام لتكون في متناول الجميع. إن آخر ما تمخض عن الجهود الكبيرة التي بذلت من مجموعة من الشركات المتخصصة، هي الحاسبات البيتية، ويمكن استخدام هذه الحاسبات من قبل الكبار والصغار، وكل حسب مستواه وبدون أية صعوبة، حيث يستطيع أي شخص كتابة برامجه بنفسه وتنفيذها دون أي صعوبة ويباع الحاسب البيتي (الشخصي) الآن في كل الأسواق كما تباع أي سلعة أخرى.

إن هذا الكتاب يوضح المعلومات الخاصة بالحاسب الإلكتروني بصورة عامة وسيأخذ الحاسب العامل بنظام (M.S.X.) نوع «صخر»، موضع التجربة والاختبار، ويشرح كيفية استخدامه وبرمجته وبأسلوب مبسط.

اللغة المستخدمة في حاسبات (M.S.X.) هي لغة البيسك (BASIC) وهذه مأخوذة من الأحرف الأولى للعبارة (Beginners Allpurpose Symbol Instruction Code) وهي لغة بسيطة جداً، وذات إمكانيات واسعة. وهذه اللغة

لا تختلف من حاسبة إلى أخرى، ولكن نتيجة للتطور السريع للحاسبات الشخصية والحاجة المستمرة لتطوير هذه اللغة لملائمة كافة الاستخدامات فظهرت بعض الاختلافات البسيطة في اللغة المستخدمة من حاسبة إلى أخرى. اتفقت بعض الشركات مع بعضها على استخدام نظام موحد، أي لغة موحدة، بحيث يمكن استخدام البرامج المعدة لنوع من الحاسبات مع النوع الآخر وكانت النتيجة هي ولادة نظام (M.S.X.)، وحاسبات «YAMAHA» «صخر» و «NEC» المتوفرة حالياً في القطر تعمل بنظام (M.S.X.)، وفي هذا الكتاب أخذنا الحاسب «صخر» كمثال لكتابة البرامج، ويمكن كذلك تنفيذ جميع البرامج على أي حاسب يعمل بنظام (M.S.X.) مباشرة. (قد تحتاج في بعض الأحيان إجراء تغيير بسيط لبعض البرامج، لتكون ملائمة للحاسب الذي تستخدمه).

استخدمت الشاشة (1) في كتابة برامج الكتاب، وهذه شاشة مخصصة للكتابة، يحتوي السطر الواحد منها على 32 حرفاً، لذلك يفضل استخدام الشاشة (1) عند طباعة البرامج وخاصة الطويلة منها، لتسهيل عملية المقارنة، وكذلك عملية اكتشاف الخطأ الذي يحصل أثناء طباعة البرنامج.

لا يسعني أخيراً إلا أن أشكر جميع الذين ساهموا في إعداد هذا الكتاب وأخص منهم السيد حيدر هاشم والسيد أركان العبيدي لمساهمتهما في توفير بعض المصادر والبرامج، وكذلك السيد بهجت عبدالواحد لإشرافه اللغوي على الكتاب.

المؤلف

رياض عزيز مرزة

الفصل الأول

تعاريف عامة

قبل الدخول بشكل تفصيلي في شرح خصائص الحاسبات، يجب أن نتعرف أولاً على بعض التعابير التي سترد كثيراً ولكنها غريبة علينا نوعاً ما.

إن الحاسب يستطيع تنفيذ البرامج من خلال جزئين مهمين فيه هما: البناء الهيكلي (HARDWARE)، والبناء البرامجي (SOFTWARE). نعني بالبناء الهيكلي، الجزء الملموس أو المادي من الحاسب. ويسمى الحاسب نفسه بهذا الاسم وكذلك كل جهاز يستخدم مع الحاسب مثل الطابعة، مسجل الكاسيت، المسجل القرصي، جهاز العرض، أو أي جهاز فيزياوي آخر. أما البناء البرامجي فهو عبارة عن الأشياء الغير فيزياوية، مثل البرامج أو المعلومات التي يمكن خزنها على الشريط المغناطيسي، أو في ذاكرة الحاسب، . . . ، والتي يمكن أن تفقد أو تتلف. وهناك نوع من البرامج تحفظ في ذاكرة خاصة في الحاسب لا تمسح أبداً حتى لو انقطعت الطاقة عنه، وهذه البرامج تسمى (FIRMWARE)، وهي التي تساعد الحاسب على إظهار العبارات التي نراها عند تشغيل الحاسب، وهي كذلك تساعد الحاسب على تشغيل جهاز التسجيل أو جهاز العرض وغيرها من الأجهزة التي تربط مع الحاسب.

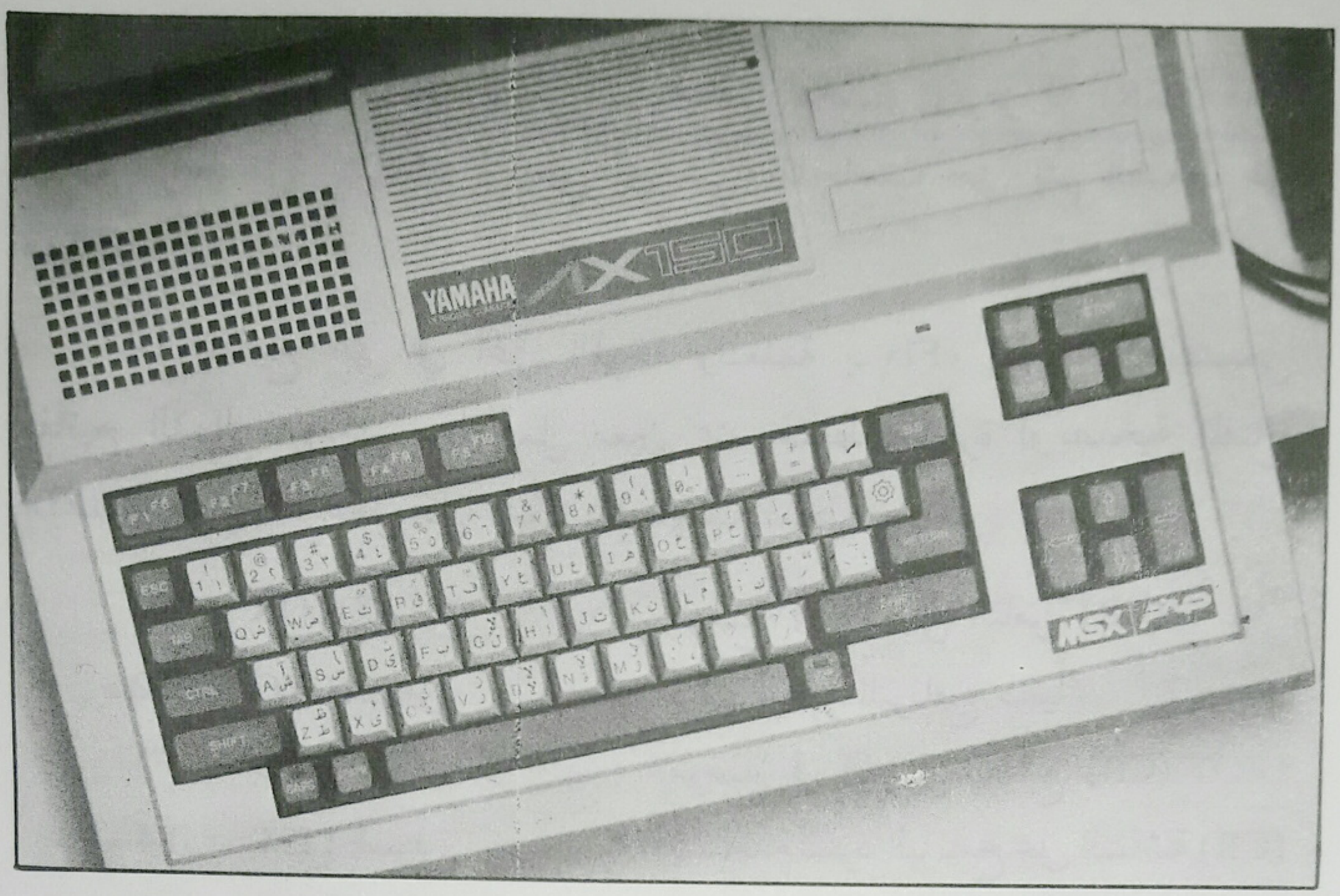
يغطي البناء الهيكلي الكثير من الأجزاء الكبيرة والمهمة في الحاسب مثل (CPU) وهذا يعني وحدة السيطرة المركزية، ويمثل العقل للحاسب، والجزء الذي يقوم بكل الأعمال، حيث يسيطر على سريان المعلومات من الذاكرة وإليها، ومن العالم الخارجي وإليه، بالإضافة إلى معالجة البيانات الداخلة إلى الحاسب. والـ (CPU) في الحاسبات البيتية ما هو إلا دائرة الكترونية متكاملة واحدة، مثل (Z80A) في حاسبات صخر و (NEC)، وكذلك في بعض الحاسبات الأخرى.

هناك جزء يعمل يداً بيد مع الـ (CPU) يدعى بالذاكرة (MEMORY) وهذا يقوم بخزن البيانات والمعلومات والبرامج، وإعادتها عند الحاجة إليها. وهناك نوعان رئيسيان من الذاكرة، الـ (RAM) وهو مختصر عبارة «Random Access Memory»، والذي يعني «ذاكرة الوصل العشوائي». وهذه تستخدم في حفظ البرامج الشخصية وبياناتها، وتمسح هذه المعلومات منها عند قطع التيار الكهربائي عنها. والنوع الآخر هو الـ (ROM) وهو مختصر عبارة «Read Only Memory» والذي يعني «ذاكرة القراءة فقط». وهذه تكون مبرمجة مسبقاً وتبقى المعلومات والبيانات مخزونة فيها بشكل ثابت. وكما يتضح لنا من أنه الجزء الذي يخزن البيانات والبرامج ويعطيها إلى الحاسب دون أن نتمكن من إضافة شيء جديد إليها. من هذا يتضح لنا أن الـ (FIRMWARE) مخزون في الـ (ROM) في الحاسبات.

الجزء الآخر هو الـ (VDU) وهو عبارة عن وحدة الشاشة وبقية الأجزاء التي تنتظم لكي تظهر المعلومات على الشاشة المرتبطة بالحاسب، وذلك ليمكن الحاسب من التحدث مع المستخدم بوضع المعلومات والبيانات على الشاشة. أما لوحة المفاتيح، فهي عبارة عن مجموعة من المفاتيح توضع أمام الحاسب كما في الطابعة الاعتيادية، وهذه تمكن المستخدم من التحدث إلى الحاسب، بطباعة الإيعازات والرموز التي يعمل الحاسب بموجبها. بعد أن أعطينا جزءاً كبيراً من التعابير التي ترد في مجال البناء الهيكلي، سوف نتطرق الآن إلى بعض التعابير التي ترد في مجال البناء البرامجي. البرامج هي مجموعة من العبارات التي تعطى إلى الحاسب وبطرق مختلفة، ليعمل الحاسب بموجبها وتشمل هذه كافة الإيعازات والبلاغات والأوامر التي تسيّر الحاسب، وهذا ما سنلاحظه مفصلاً في الفصول القادمة.

*

**



لوحة المفاتيح في الحاسب YAMAHA 150 (صخر)

ولا يستمر في العمل إلا بعد إعطائه إيعازاً بذلك.

المفاتيح الأربعة المؤشرة بالأسهم تستخدم في تحريك «المشيرة» حسب اتجاه السهم، دون طبع أي رمز.

المفتاح «DEL» يستخدم لمسح الرمز الذي تقف المشيرة عليه. والمفتاح «INS» فيستخدم لإدخال رمز معين داخل الكلمة، فعند الضغط عليه تصغر «المشيرة» إلى النصف، لتبين لنا أننا في وضع الإدخال. أما المفتاح «BS» فيستخدم لمسح الرمز الذي يقع على يمين «المشيرة»:

المفتاح «CLS/HOME» له تأثيران فإذا ما ضغط عليه فإن «المشيرة» ستتحرك إلى الزاوية العليا اليسرى من الشاشة دون مسح ما مطبوع على الشاشة. أما إذا ضغط عليه مع المفتاح «SHIFT» فذلك يؤدي إلى مسح الشاشة مما مطبوع عليها تماماً.

المفتاح «RETURN» يسمى بـ «عتلة الإدخال» ويرمز له (CR). عند طباعة أمر ما على الشاشة، لا يعني أي شيء للحاسب حتى يتم الضغط على هذا المفتاح.

أما المفاتيح التي في أعلى اللوحة والمعلمة بـ «F1» إلى «F10» فتسمى مفاتيح الدوال، ولكل منها عمل معين عند ضغط مباشرة أو بصحبة المفتاح «SHIFT». وأدناه عمل كل منها: (للحاسب صخر)

F1	COLOR لون.	موضح في الفصل الثامن (CR)
F2	AUTO ذاتي.	يرقم أسطر البرنامج ذاتياً
F3	GOTO اقصد.	موضح في الفصل السابع (CR)
F4	LIST اسرد.	يستخدم لسرد البرنامج على الشاشة (CR)
F5	RUN نفذ.	عند الضغط عليه يتم تنفيذ البرنامج
F6	COLOR_ _ لون.	يرجع اللون الأساسي للشاشة
F7	CLOAD كاحمل.	لتحميل برنامج من شريط الكاسيت (CR)
F8	CONT استمر.	لاستمرار البرنامج بعد أن تم إيقافه
F9	LIST_ _ اسرد.	موضح في الفصل السادس
F10	CLS + RUN امسح + نفذ.	يمسح الشاشة ثم ينفذ البرنامج

الفصل الثالث تشغيل الحاسب

أول خطوة في منهاج التعليم على استعمال الحاسب الشخصي هي التعلم على كيفية ربط الحاسب مع الأجهزة الأخرى ومن ثم تشغيله.

كما نلاحظ في الشكل يمكن ربط الحاسب بجهاز عرض خاص به (MONITOR)، وإذا لم يتوفر لدينا هذا الجهاز يمكن لنا استخدام التلفزيون المنزلي لذلك. ويمكن لنا كذلك ربط الحاسب بجهاز التسجيل لتسجيل البرامج التي نرغب بالاحتفاظ بها، وذلك باستخدام الأسلاك الخاصة والمرفقة مع الحاسب، مع الاستعانة بالكتيب الخاص بجهاز التسجيل. كذلك يمكن لنا ربط الطابعة السطرية وعصا التحكم إذا ما توفرت.

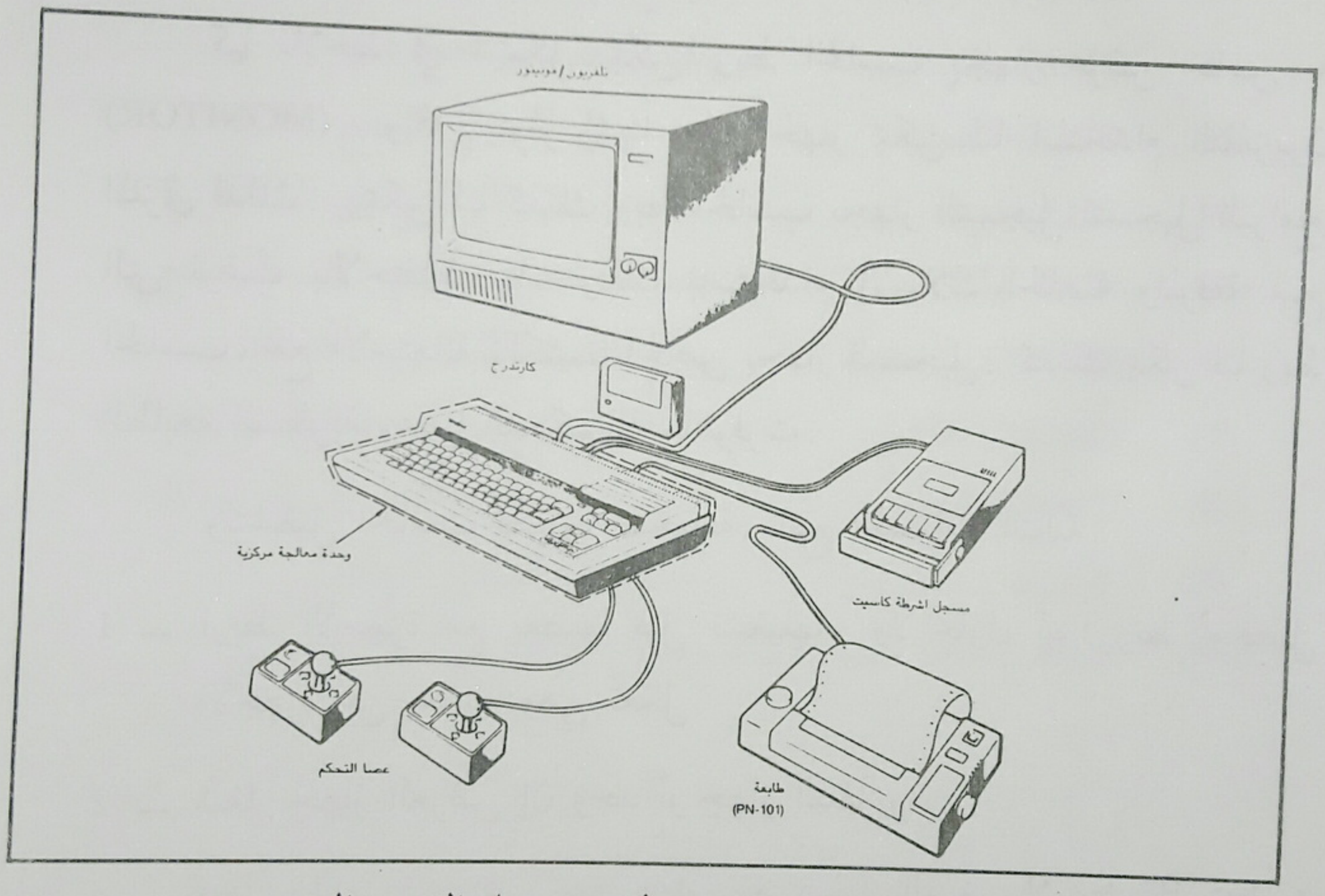
ولتشغيل الحاسب بصورة صحيحة، اتبع الخطوات التالية:

1 ـ اربط الأجهزة مع بعضها قبل تشغيلها، ولا تحاول أبداً ربط أو فصل الأجهزة عن بعضها وهي تعمل.

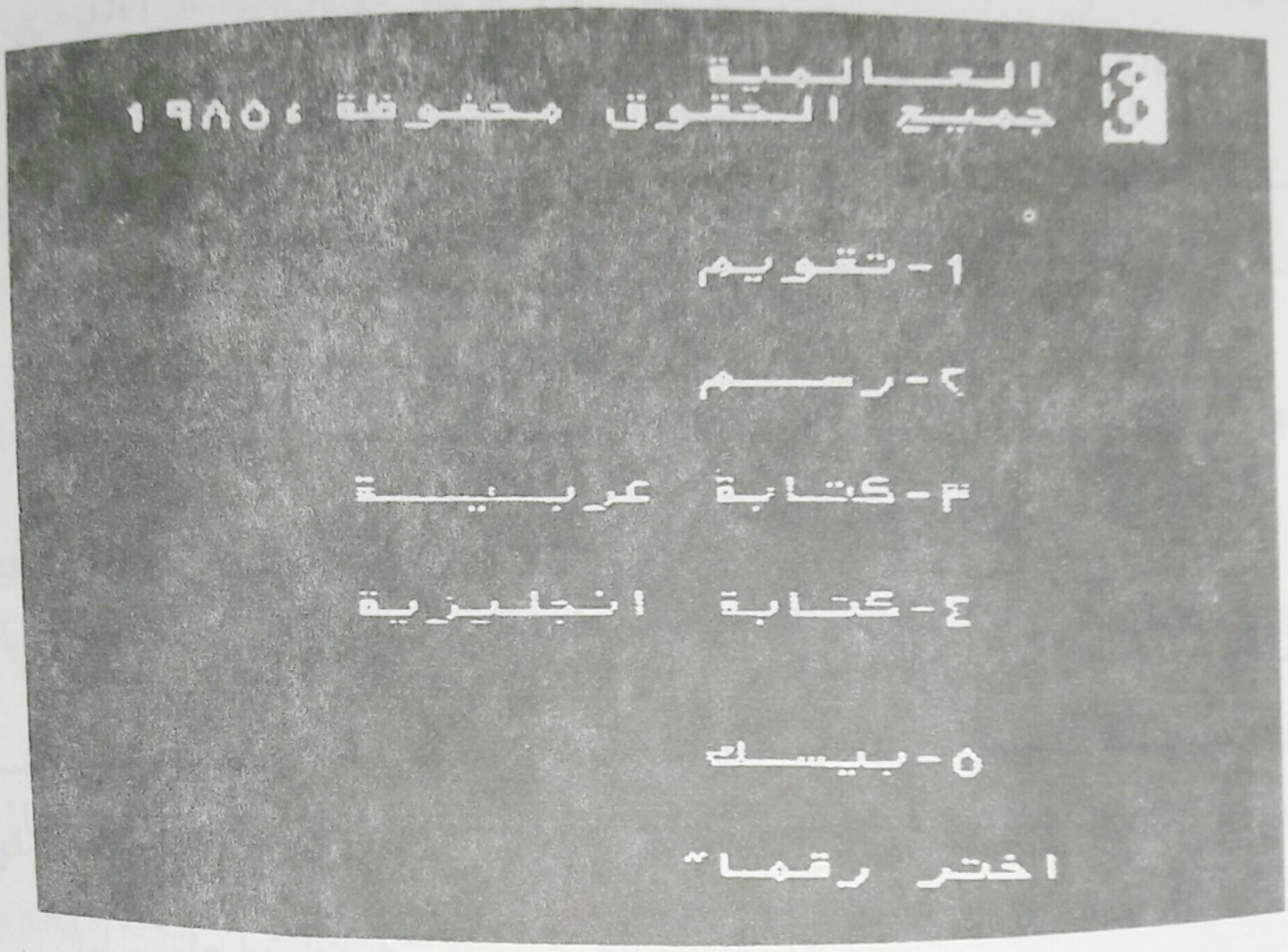
2 ـ شغل جهاز العرض إن وجد أو جهاز التلفزيون.

3 ـ شغل الحاسب، ونظم جهاز التلفزيون (جهاز العرض لا يحتاج إلى تنظيم) على أحد قنوات الـ (UHF)، وعند حصول حالة التنظيم سترى على الشاشة المعلومات الخاصة بالحاسب (في حالة استخدام جهاز «صخر» العربي ستظهر لك قائمة الاختيارات، اضغط على الرقم «5» للانتقال إلى طور البرمجة).

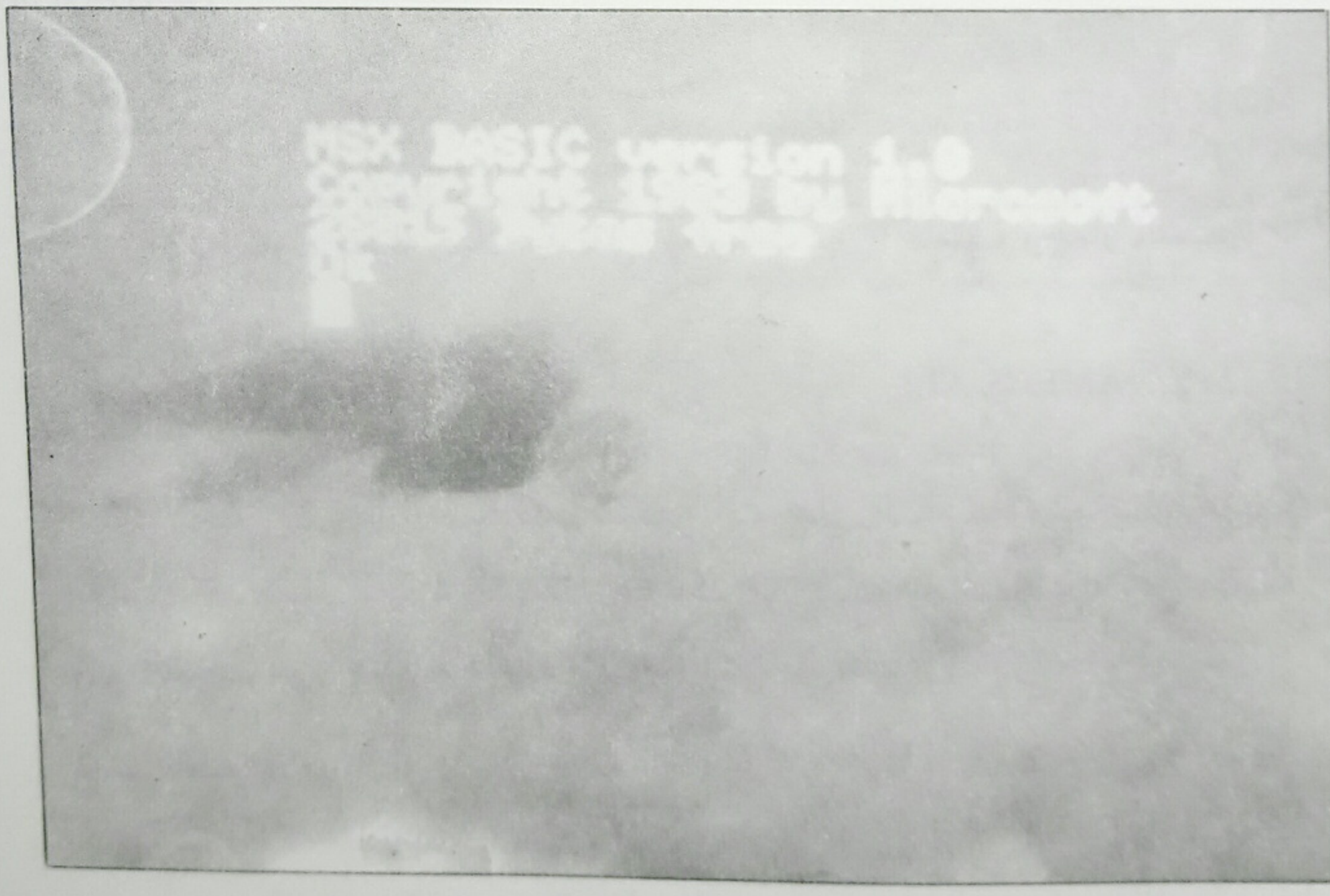
4 ـ عند استخدام الحاسب «صخر» يمكن لك الانتقال مباشرة إلى طور البرمجة (استخدام اللغة الانكليزية فقط)، وذلك بالضغط على المفاتيح «SHIFT»



الحاسب YAMAHA (صخر) مرتبط مع مسجل القرص السريع Q.D
ومسجل كاسيت وطابعة، إضافة إلى جهاز العرض



الشاشة الخاصة بـ «صخر»



الشاشة الخاصة بـ «MSX»

و «CTRL» و «ESC» سوية ومن ثم تشغيل الحاسب مع الاستمرار بالضغط على المفاتيح الثلاثة إلى أن تظهر المعلومات الخاصة بالحاسب.

5 ـ اضغط على المفاتيح «A» إلى «Z» ستشاهدها على الشاشة، لاحظ لو أنك ضغطت على أحد المفاتيح بصورة مستمرة، هذا يؤدي إلى طبع الحرف المقابل لذلك المفتاح وبصورة متكررة.

فحص جهاز التسجيل:

لتوصيل جهاز التسجيل مع الحاسب راجع التعليمات الخاصة بجهاز التسجيل والحاسب. وللتأكد من صحة الربط استخدم الحاسب للسيطرة على جهاز التسجيل، وكالآتي:

1 ـ اطبع العبارة التالية:

```
MOTOR ON
(CR)
```

ستلاحظ أن الشريط في جهاز التسجيل أخذ بالدوران.

2 ـ اطبع العبارة التالية:

```
MOTOR OFF
(CR)
```

الآن حاول معرفة صلاحية نقل المعلومات بين الحاسب وجهاز التسجيل.

1 ـ اطبع العبارة التالية:

```
10 MY NAME IS ALI
(CR)
```

لاحظ أن عبارة «O.K.» ستظهر لك على الشاشة وهذه تدل على أن الحاسب قد تقبل العبارة السابقة. وللتأكد من ذلك يمكنك طبع الأمر «LIST» وثم الضغط على عتلة الادخال (CR)، ستلاحظ ظهور العبارة على الشاشة.

2 ـ اطبع الأمر التالي:

```
CSAVE «A1»
```

وقبل أن تضغط على عتلة الادخال، جهز جهاز التسجيل وذلك بالضغط

على مفتاحي التسجيل (PLAY + REC). اضغط الآن على عتلة الادخال (CR) ستلاحظ دوران الشريط لثوانٍ ثم توقفه وظهور عبارة «O.K.».

3 ـ اضغط على مفتاح الإيقاف «STOP» في جهاز التسجيل، ثم مفتاح إعادة الشريط إلى البداية «REW».

4 ـ اطبع الآن الأمر «MOTOR ON» واضغط على عتلة الادخال (CR)، لتشغيل محرك جهاز التسجيل.

5 ـ اطبع الأمر «MOTOR OFF» وضغط على (CR) حال انتهاء إعادة الشريط.

6 ـ اضغط على المفتاح «PLAY» في جهاز التسجيل. ثم اطبع الأمر التالي:

```
CLOAD «A1»
(CR)
```

ستلاحظ دوران الشريط ثم توقفه بعد ثوانٍ وظهور العبارة التالية:

```
FOUND A1
O.K.
```

وتوقف الجهاز عن الدوران. وهذا يعني أن العبارة السابقة قد انتقلت من جهاز التسجيل إلى الحاسب. وإن لم تتم عملية النقل بصورة صحيحة فإن الحاسب سيظهر لك العبارة التالية:

```
I/O  Error
```

لذلك أجرِ الخطوات التالية:

1 ـ افحص الربط بين الحاسب وجهاز التسجيل، وتأكد من أن الأسلاك موصلة بالشكل الصحيح.

2 ـ غيّر مستوى ارتفاع الصوت.

3 ـ افحص رأس التسجيل (HEAD)، ونظفه باستخدام قطعة صغيرة من القطن مبللة قليل من الكحول.

مسجل القرص السريع (Q.D.) :
(مع حاسبات «YAMAHA صخر» فقط) :

لا يختلف عمل مسجل القرص السريع عن عمل جهاز التسجيل سوى كون الأول أسرع لإستخدامه قرص ممغنط بدلاً من الشريط. ولربط الجهاز مع الحاسب اتبع الخطوات التالية :

1 ـ اقطع الطاقة الكهربائية عن الحاسب.
2 ـ ادخل مقبس التوصيل في المكان المخصص له في أعلى الحاسب.
3 ـ أوصل الجهاز بالطاقة الكهربائية.
4 ـ شغّل الحاسب.

للقرص السريع أوامر خاصة به يمكن استخدامها لحفظ وتحميل البرامج والمعلومات. وتخصص قسم من مفاتيح الدوال لهذه الأوامر عند ربط الجهاز بالحاسب. وهذه الأوامر هي :

ربط مسجل القرص السريع (Q.D) مع الحاسب

ملاحظة: قبل كل أمر من أوامر القرص السريع يستخدم الأمر «CALL» ويمكن الاستعاضة عنه بعلامة الناقص «-».

*** QDFORMAT

يستخدم هذا الأمر لتنظيم وتنظيف القرص الجديد، وكذلك لمسح ما هو موجود على القرص بالكامل. ويستخدم بالصيغة التالية:

-QDFORMAT

ليظهر على الشاشة الاستفسار التالي:

ARE YOU SURE (Y/N)

اضغط على مفتاح الرمز «Y» والتي تعني «نعم». وبذلك تظهر العبارة التالية:

COMPLETE

والتي تعني أن العملية انتهت.

*** SAVE

يستخدم هذا الأمر في عملية حفظ البرامج، ويكون بالصيغة التالية:

-SAVE («QD : اسم البرنامج»)

*** LOAD

يستخدم هذا الأمر لتحميل برنامج من القرص إلى الحاسب، ويكون بالصيغة التالية:

-LOAD («QD : اسم البرنامج»)

*** QDFILES

يستخدم هذا الأمر لعرض أسماء البرامج المحفوظة على القرص. ويستخدم بالصيغة التالية:

-QDFILES

*** QDKILL

يستخدم هذا الأمر لمسح البرنامج الأخير من على القرص، ويكـون بالصيغة التالية:

-QDKILL («اسم البرنامج»)

*** QDKEY

يستخدم هذا الأمر عند الحاجة إلى إرجاع مفاتيح الدوال إلى الدوال الأساسية المثبتة في الحاسب وليس تلك الخاصة بالقرص السريع. وعند إعطاء الأمر مرة أخرى إلى الحاسب، ترجع الدوال الخاصة بالقرص السريع وصيغة الأمر تكون كالآتي:

-QDKEY

بعد أن تمكنا من تشغيل الحاسب والأجهزة أعلاه وبصورة صحيحة، سوف لن نواجه أية صعوبة في عملية الربط والتشغيـل للأجهـزة الملحقة الأخرى، مع ضرورة الرجوع إلى الكتيبات الخاصة بها والمرفقة معها.

الفصل الرابع — الطور المباشر

قبل البدء بهذا الفصل، يجب أن نراجع معلوماتنا في استخدام لوحة المفاتيح (الفصل الثاني). في هذا الفصل سوف نتعرف على أحد الإيعازات المهمة في لغة البيسك، وهو الأمر «اطبع PRINT». هذا الأمر هو المسؤول عن إظهار النتائج على الشاشة، وبدونه لا يمكن لنا معرفة النتائج التي توصل إليها الحاسب. فلنحسب مثلاً ناتج (2 + 3). سنوجه أمراً إلى الحاسب لطبع ناتج الجمع، ويكون الأمر بالصيغة التالية:

اطبع 2 + 3 — PRINT 2 + 3

وهذا يعني اطبع الجواب على الشاشة، فبعد أن تضغط على مفتاح الإدخال (CR) ستشاهد الرقم (5) ثم العبارة «O.K» والتي تعني أن الحاسب مستعد لتقبل أوامر أخرى.

مثـال:

احسب ناتج 5 − 3 + 4

الإيعاز يكون كالآتي:

```
PRINT  5 − 3 + 4
(CR)
6
O.K.
```

أما إذا كانت المعادلة غير قابلة للحل فإن الحاسب يعطي العبارة التالية:

```
Syntax error
```

والتي تعني وجود خطأ نحوي.

الآن اطبع المثال الأول وبالصورة التالية:

```
PRINT «2 + 3»
(CR)
```

لاحظ أن الحاسب سيطبع التالي:

```
2+ 3
O.K
```

هذا يدل على أن الحاسب يطبع ما محصور بين الحاصرتين العلويتين دون إجراء أي عملية حسابية عليها. وتستخدم هذه الطريقة أيضاً عند طبع الجمل.

للتعرف على خصائص الأمر بصورة أفضل، حاول تنفيذ الأمثلة التالية:

```
PRINT 3 + 2 ^ 2 (CR)                    («^» علامة الأس)
7
O.K
PRINT 3 ^ 2 + 2 (CR)
11
O.K
PRINT 5* 4 + 2 (CR)                     («*» علامة الضرب)
22
O.K
PRINT 5 * (4 + 2) (CR)
30
O.K
PRINT 10 / 2 * 5 + 2 (CR)
3
O.K
PRINT (10 / 2) * (5 + 2) (CR)
35
O.K
PRINT «MY NAME IS ALI» (CR)
MY NAME IS ALI
O.K
PRINT MY NAME IS ALI (CR)
Syntax error
O.K
```

تستخدم الفوارز مع الأمر «PRINT» ولكل منها تأثير مختلف، فإن استعملنا الفارزة الاعتيادية (،) فإن العبارة بعد الفارزة ستطبع على بعد 8 فراغات.

```
PRINT 3,5 (CR)
3               5
O.K
```

```
PRINT 3,8,5 (CR)
3                    5                    8
O.K
PRINT «2 + 3=», 2 + 3  (CR)
2+ 3 =                    5
O.K
```

أما الفارزة المنقوطة (؛) فإن الحاسب لن يترك أي فراغ عند الطبع.

حاول الأمثلة السابقة مع إبدال الفوارز بفوارز منقوطة، ستلاحظ التالي:

```
PRINT 3;5  (CR)
35
O.K
PRINT 3;5;8  (CR)
358
O.K
PRINT «2 + 3 =»; 2 + 3  (CR)
2 + 3 = 5
O.K
```

*
**

ملاحظة:

اينما وردت العلامة (« ») تعنى حاصرتين علويتين (" ").

الفصل الخامس تصحيح الأخطاء على الشاشة

عملية تصحيح الأخطاء الناتجة أثناء طباعة البرامج تسّمى بعملية المونتاج (EDITING). والمفاتيح المستخدمة لذلك هي:

- المفاتيح المسيطرة على حركة المشير «CURSOR» (المفاتيح الأربعة المعلمة بالأسهم).
- مفتاحا المسح «DEL» والإضافة «INS».
- مفتاح إدخال «RETURN» (CR).

كما بينا سابقاً بأن أي جملة أو فقرة لا يمكن للحاسب أن يتفهمها إلا بعد الضغط على مفتاح الإدخال (CR)، ويجب أن نتذكر دائماً بأن الحاسب لا يتعرف على أي أمر أو عبارة ما لم نستخدم المفتاح (CR).

إن نظام المونتاج في حاسبات (M.S.X.) يسمى بالنظام المباشر، حيث يمكن إجراء عملية المونتاج مباشرة على الشاشة وذلك بتصحيح الخطأ دون الحاجة إلى إعادة كتابة العبارة ككل، وهذا ما سنراه في هذا الفصل.

في بعض الأحيان وبعد أن تتم كتابة البرنامج وتغذيته إلى الحاسب، وحين التنفيذ بإعطاء الأمر «RUN»، تظهر على الشاشة عبارة توضح لنا وجود خطأ في البرنامج المنفذ ولذا يجب إجراء عملية التصحيح.

إن العبارات التي يصدرها الحاسب للدلالة على نوع الخطأ كثيرة وسندرسها في الفصل القادم (الفصل السادس). أما الآن فسنتعلم عملية التصحيح بصورة عامة.

اطبع الآن العبارة التالية (مع ما فيها من خطأ) كمثال لإجراء عملية التصحيح.

PRENT «THIS IS AN EDITING TEST»

ثم اضغط على مفتاح الإدخال، حينها ستظهر لك على الشاشة العبارة التالية:

Syntax error

وهذه تعني وجود خطأ لغوي في مضمون العبارة.

لنحاول الآن إجراء عملية المونتاج وفق الخطوات التالية:

- حدد طبيعة الخطأ وحاول اكتشافه.
- بالتأكيد فإن الخطأ في العبارة السابقة هو في الأمر «PRENT» حيث أن الصيغة الصحيحة هي «PRINT».
- استخدم مفاتيح السيطرة على المشيرة، لتنقلها إلى مكان الخطأ، أي عند الحرف «E».
- أجرِ عملية التصحيح في العبارة، وذلك بطبع الحرف «I»، حيث سيقوم الحاسب بوضعه مباشرة مكان الحرف «E»، وبذلك يتم تصحيح إملاء الأمر. وبذلك تصبح العبارة السابقة كالآتي:

PRINT «THIS IS AN EDITING TEST»

- اضغط على مفتاح الإدخال (CR).

ستظهر لك الآن العبارة التالية على الشاشة:

THIS IS AN EDITING TEST

لاحظ أن الخطأ قد تم تصحيحه وأن الحاسب قد تقبل العبارة بعد إجراء التصحيح عليها. المثال السابق هو من الأمثلة البسيطة، ولكن ماذا يمكن لنا أن نفعل لو كان الخطأ على الشكل التالي:

PRINNT «THIS IS AN EDITING TEST»

بعد الضغط على مفتاح الإدخال، نلاحظ أن الحاسب يعطي إشارة الخطأ السابقة ذاتها، ماذا سنفعل؟ سنقوم بعملية المونتاج بالخطوات السابقة نفسها، ولكن خطوة التصحيح تكون باستخدام المفتاح «DEL» الذي يعني «امسح»، ضع المشيرة فوق الحرف «N» قبل الضغط على المفتاح «DEL».

الآن وبعد الضغط على مفتاح المسح، نلاحظ اختفاء الحرف «N»، وأصبحت العبارة كالآتي:

PRINT «THIS IS AN EDITING TEST»

اضغط الآن على مفتاح الإدخال (CR) لتلاحظ أن الحاسب سوف يطبع العبارة التالية:

THIS IS AN EDITING TEST

اطبع الآن العبارة التالية:

PINT «THIS IS AN EDITING TEST»

ستلاحظ بعد الضغط على مفتاح الإدخال ظهور عبارة تدل على وجود خطأ لغوي في العبارة. والخطأ هو عدم وجود الحرف «R» في الأمر «PRINT». إذن ما نحتاجه الآن هو إدخال الحرف «R» في الأمر السابق، ويتم ذلك كالآتي:

- حرك المشيرة، باستخدام مفاتيح السيطرة على حركة المشيرة، لتضع المشيرة فوق الحرف «I» في كلمة «PINT».
- اضغط على المفتاح «INS»، وبذلك ستلاحظ أن المشيرة ستصغر إلى النصف وهذا يعني أن الحاسب سيعمل في «طور الإضافة».
- اضغط على الحرف «R» وستلاحظ أن الحرف «R» سيجعل العبارة تزحف إلى اليمين وتصبح كالآتي:

PRINT «THIS IS AN EDITING TEST»

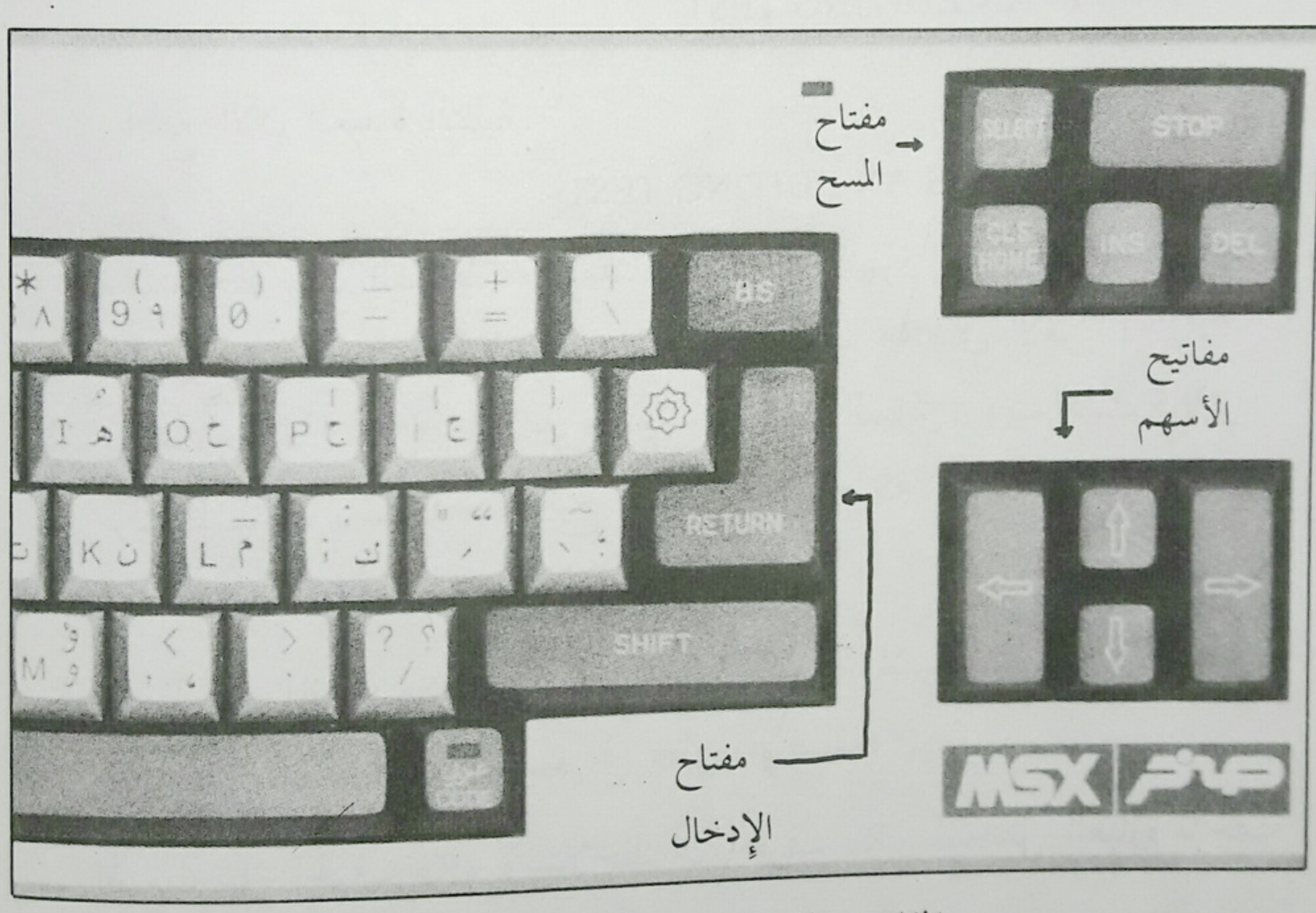
- اضغط على المفتاح «INS» مرة أخرى، ستلاحظ رجوع المشيرة إلى حجمها الطبيعي.
- اضغط على مفتاح الإدخال (CR).

سيطبع لك الحاسب العبارة التالية على الشاشة:

THIS IS AN EDITING TEST

مما سبق تعلمنا المبادىء الأساسية لعملية المونتاج للبرامج، وأخيراً يجب أن لا ننسى النقاط التالية:

- لا تكون عملية التصحيح مقبولة بالنسبة للحاسب ما لم تدخل العبارة الصحيحة إلى الحاسب باستخدام مفتاح الإدخال (CR).
- بعد إجراء عملية التصحيح، والضغط على مفتاح الإدخال، تظهر عبارة «O.K» على الشاشة، لتدل على أن العملية كانت صحيحة.



المفاتيح الخاصة بعملية المونتاج (التصحيح)

الفصل السادس

رسائل الخطأ

عند استخدامنا للحاسب نقع في بعض الأخطاء، وإذا لم يكشف عنها الحاسب وينبهنا عنها، فإن النتائج التي يعطيها الحاسب تكون خاطئة.

لهذا فإن مصنعي الحاسبات قد وضعوا في ذاكرة الحاسب الثابتة (ROM) برامج خاصة وظيفتها الأساسية الكشف عن الأخطاء. وقد لاحظنا ذلك في الفصل السابق. وللحاسب إمكانية الكشف عن الخطأ وتحديد نوعه ومكانه، ولكنها لا تستطيع تحديد إلا الأخطاء اللغوية والمنطقية والأخطاء المؤثرة في سير تنفيذ البرنامج. أما الأخطاءالأخرى من إعطاء أرقام خاطئة أو معادلة خاطئة، فإن الحاسب لن يستطيع الكشف عنها.

لو أننا طبعنا المعادلة التالية:

```
X = Y^2 + 4
```

بالشكل التالي:

```
X = Y + 4
```

فإن الحاسب لا يعتبر هذا خطأً، حيث المعادلة المطبوعة صحيحة، ولا يستطيع أن يحدد المعادلة المطلوب طباعتها. أما إذا طبعت المعادلة بالشكل التالي:

```
Y ^ 2 + 4 = X
```

فإن الحاسب يعطي إشارة بالخطأ، لأن الحاسب يعتمد على وجود متغير واحد فقط في الجانب الأيسر من المعادلة.

إذا كانِ الخطأ قد حصل في الطور المباشر، فإن الخطأ سيظهر بصورة مباشرة بإظهار العبارة:

```
Syntax error
```

أما إذا كان الخطأ قد حصل خلال تنفيذ البرنامج، فإن العبارة السابقة تكون غير كافية، وبذلك يحدد الحاسب نوع الخطأ والسطر الذي حدث الخطأ فيه. مثلاً:

Syntax error in line No. 100

هذا يدل على وجود الخطأ في السطر 100، لذلك فإن تصحيح الخطأ سيكون بمراجعة السطر 100 فقط وتصحيحه. ولكل خطأ رمز معين في الحاسب يمكن الرجوع إليه أو استخدامه مع الايعاز «..ON ERROR». وأدناه عبارات الخطأ والرموز الخاصة بها:

1 Next without for

تالي بدون من

تظهر هذه العبارة عند وجود «NEXT» دون وجود «FOR»، أو وجودها بشكل خاطىء. مثلاً:

```
FOR X = 1 TO 10
NEXT Y
```

لاحظ بأن المتغير مع «FOR» هو «X»، أما الموجود مع «NEXT» فهو «Y».

2 Syntax error

خطأ نحوي

تظهر هذه العبارة عند إدخال إيعاز غير مفهوم إلى الحاسب، كان يكون فيه خطأ إملائي، أو عدم وجود بعض الفوارز الضرورية، أو استخدام الحرف «O» بدلاً من الصفر «0» أو العكس. وكذلك عند وجود أقواس ناقصة. لاحظ الأمثلة التالية:

FOR I = 1 TO 2O استخدم حرف الـ «O» مكان الـ «0»

يوجد قوسان من جهة وقوس واحد من الجهة المقابلة

. A = INT (20.3))

PRINT HELLO» عدم وجود الحاصرة العلوية من الجهة اليمنى

3 RETURN WITHOUT GOSUB ارجع دون تفرع

يعطي الحاسب هذا الخطأ عند وجود الايعاز «RETURN» دون وجود البلاغ «GOSUB»

4 Out of data بيان ناقص

تظهر هذه العبارة عندما لا توجد بيانات كافية بقدر ما يحتاجه الحاسب أي بقدر المتغيرات الموجودة في الخطوة التي تحمل الايعاز «READ» مثلاً:

```
READ A, B, C
DATA 15, 3
```

5 Illegel function call نداء دالة غير مشروع

تظهر هذه العبارة عندما تكون النتيجة الحسابية لمعادلة رياضية أكبر من الحد الذي يتحمله الحاسب. أو طلب دالة رياضية لعدد غير واقعي، مثلاً:

- اللوغاريتم للصفر، أو لرقم سالب.
- الجذر التربيعي لرقم سالب.
- طلب الزاوية التي جيبها أكبر من «1»
- استخدام الرسوم والأشباح مع الشاشة «0»
- استخدام رموز مع الايعاز «DRAW» والبلاغ «PLAY» غير الرموز المخصصة لهما.

6 Over flow زائد عن الحد

تظهر هذه العبارة عند استخدام رقم كبير جداً مع الحاسب، بحيث لا يتمكن من خزنه أو إظهاره على الشاشة.

7 Out of memory الذاكرة غير كافية

تظهر هذه العبارة عندما يدخل إلى الحاسب برنامج أكبر من سعة الذاكرة، حيث يعجز الحاسب عن خزنه ومن ثم تنفيذه. يمكنك استخدام البلاغ «FRE(0)» لمعرفة الحجم المتبقي من الذاكرة. وكذلك يمكن مسح البيانات السابقة باستخدام البلاغ «CLEAR».

8 Undefined line number سطر غير محدد

هناك بعض الإيعازات تقود الحاسب إلى سطر معين من البرنامج، لذلك يجب أن يكون السطر المحدد فيها موجود ضمن البرنامج وإلا ستظهر عبارة الخطأ هذه.

9 Subscript out of rang خارج نطاق المصفوفة

تظهر هذه العبارة عند تجاوز عدد المتغيرات عن البعد الذي حدد لها ضمن الإيعاز «DIM». مثلاً:

```
DIM A(100)
A(102) = 12
```

هذا خطأ حيث حدود المتغير هي من A(0) إلى A(100)

```
DIM A(6,4)
A(6,5) = 7
```

هذا خطأ أيضاً حيث حدود المتغير هي من A(0,0) إلى A(6,4) وبذلك يكون المتغير A(6,5) خارج النطاق.

10 Redimansional array بعد مكرر لمصفوفة

تظهر هذه العبارة عندما يكرر إعطاء بعد معين لمتغير مرتين، أي ظهور متغير واحد مرتين وفي خطوتين تحملان الإيعاز «DIM». مثلاً:

```
DIM A(20)
DIM A(10)
```

11 Division by zero القسمة على صفر

تظهر هذه العبارة عند وجود خطوة حسابية فيها المقام (0).

12 Illegale direct أمر مباشر غير مشروع

تظهر هذه العبارة عند استخدام أوامر خاطئة وفي الطور المباشر.

13 Type mismatch

عدم توافق النوع

تظهر هذه العبارة عند وجود عدم توافق ضمن البرنامج. مثلاً تخصيص حزمة من الرموز لمتغير عددي.

14 Out of string space

حيز خارج نطاق الحزمة

تظهر هذه العبارة عندما تكون حزمة المتغيرات تتجاوز مقدار المتبقي من الذاكرة أو المجال الذي حدد بالبلاغ «CLEAR».

19 Device I/O error

خطأ في جهاز الإدخال والإخراج (د / خ)

تظهر هذه العبارة عند وجود خطأ في تشغيل جهاز الإدخال / الإخراج.

56 Bad file name

خطأ اسم الملف

تظهر هذه العبارة عند استخدام اسم ملف طويل جداً مع أحد عبارات الادخال أو الإخراج.

وهناك رسائل خطأ كثيرة أخرى، لكنها نادراً ما تظهر، ويمكن الرجوع إلى الكتيب الخاص بالحاسب عند ظهورها.

*
* *

الفصل السابع مبادىء البرمجة

في هذا الفصل سنبدأ بدراسة مبادىء البرمجة لحاسبات (M.S.X.) وذلك من خلال البرامج التي سترد فيها والتوضيحات لكل خطوة من خطواتها.

تكتب البرامج بلغة نفهمها نحن وكذلك الحاسب، الذي يحولها بدوره إلى لغة الآلة (MACHINE CODE). وتحتوي هذه اللغة على أوامر ودوال وبلاغات يفهمها الحاسب، ولكن لا يقوم بمعالجتها إلا بعد طلب ذلك منه، ويتم هذا بواسطة الأمر «RUN» والذي يعني نفذ، وذلك بطباعة كلمة «RUN» والضغط على مفتاح الإدخال (CR).

تدخل المعلومات إلى الحاسب بطريقتين: الأولى في الطريقة المباشرة وهي ما لاحظناه عند طباعة الأمر «RUN» والضغط على (CR)، حيث تقبل الحاسب ذلك وأجرى عملية التنفيذ. أما الطريقة الثانية فهي الطريقة غير المباشرة، حيث تكتب الخطوات بصورة متسلسلة وبأسطر مرقمة، وهذه تكوّن البرنامج، ومن ثم يطلب من الحاسب تنفيذ هذا البرنامج.

إذن من متطلبات البرنامج الأساسية هي كتابته بأسطر مرقمة بأرقام متسلسلة، وهذه تسمى بأرقام السطور (LINE NUMBERS). ويمكن للسطر الواحد أن يحتوي على خطوة واحدة أو أكثر. لنبدأ بالبرنامج البسيط التالي، وقبل طباعته اضغط على المفتاحين (CLS/HOME + SHIFT) لتنظيف الشاشة.

```
10  PRINT «MY NAME IS»
20  PRINT «ALI»

RUN
MY NAME IS
ALI
O.K
```

وهنا يجب أن لا ننسى أن نضغط على مفتاح الإدخال (CR) في نهاية كل سطر لغرض إدخاله إلى الحاسب.

ينفذ البرنامج بصورة متسلسلة حيث ينفذ السطر 10، ثم 20، وثم 30 وهكذا وتكون أرقام الأسطر بفارق 10، وهذه تفيدنا عند الحاجة إلى إضافة سطر بين أسطر البرنامج بعد الانتهاء من طباعته.

امسح الشاشة مرة أخرى، واطبع كلمة «LIST اسرد» واضغط على (CR)، لاحظ أن الحاسب سيسرد البرنامج على الشاشة، هذا يوضح أن مسح الشاشة لا يؤدي إلى مسح البرنامج من الحاسب. ولغرض مسح البرنامج من الحاسب، اطبع الأمر «NEW جديد» واضغط على (CR). فإذا ما طلبت الآن من الحاسب سرد البرنامج، ستظهر لك عبارة «O.K تم» فقط مما يدل على أن البرنامج قد مسح من ذاكرة الحاسب.

لنجرب الآن برنامجاً أكبر من السابق. اطبع البرنامج أدناه ولا تنسَ الضغط على (CR) في نهاية كل سطر.

```
10    PRINT «THIS IS THE FIRST TEST PROGRAM»
20    PRINT
30    PRINT «THIS PROGRAM WILL ADD THREE NUMBERS»
40    PRINT «THE NUMBERS ARE 100, 200, AND 55»
50    PRINT
60    PRINT «100 + 200 + 55 =»; 100 + 200 + 55
70    END
```

لاحظ البلاغ «END نهاية» في السطر 70 الذي يخبر الحاسب بانتهاء البرنامج. وإذا ما كانت نهاية البرنامج في السطر الأخير منه يمكن الاستغناء عن كتابة البلاغ.

إذا ما حصل وأن أخطأت في كتابة أي سطر من أسطر البرنامج، أجرِ عملية التصحيح (المونتاج) على ذلك السطر ولا تنسَ أن تضغط على (CR) بعد الانتهاء من العملية لكل سطر، لإدخال التصحيح إلى الذاكرة (راجع الفصل الخامس).

بعد أن أدخلت البرنامج إلى الحاسب، اطبع الأمر «RUN» لتنفيذ البرنامج ولاحظ أن الحاسب سيطبع لك على الشاشة:

```
RUN
THIS IS THE FIRST TEST PROGRAM
THIS PROGRAM WILL ADD THREE NUMBERS
THE NUMBERS ARE 100, 200 AND 55
100 + 200 + 55 = 355
```

بعد أن تم تنفيذ البرنامج، اضغط على المفتاح «F5» (أحد مفاتيح الدوال) ولاحظ أن البرنامج ينفذ مرة أخرى، من هذا نستنتج أن المفتاح «F5» يعمل كما لو كتبت الأمر «RUN» وضغطت على (CR). اضغط الآن على المفتاح «F4»، لاحظ أن كلمة «LIST» تظهر على الشاشة، ثم اضغط على (CR) ولاحظ أن الحاسب سيسرد البرنامج على الشاشة. إذا ما كانت هناك طابعة سطرية مرتبطة بالحاسب، اطبع كلمة «LLIST» واضغط على (CR)، لاحظ أن هذا يؤدي إلى سرد البرنامج على الطابعة. حاول الآن تعديل البرنامج السابق بتغيير كل البلاغات «PRINT» إلى «LPRINT»، ستلاحظ بأن النتائج ستظهر على الطابعة. من البلاغات المهمة والمفيدة هو البلاغ «REM ملحوظة» وهذا البلاغ يسمح لنا بوضع ملحوظات داخل البرنامج لتسهل علينا متابعته وقراءته، دون التأثير على عمله، حيث كل ما يرد بعد البلاغ «REM» في الخطوة لا يتعامل معه الحاسب.

أضف السطر التالي إلى البرنامج السابق، ثم نفذ البرنامج.

```
5   REM THIS IS A REMINDER (CR)
```

عندها لن تلاحظ أي تغيير في الناتج.

سوف نستخدم هذا البلاغ بكثرة في هذا الفصل والفصول القادمة لإيضاح عمل البرامج التي سترد فيها. بإمكانك حذف كافة الخطوات الحاوية على البلاغ لاختصار البرامج دون التأثير عليها.

*** المتغيرات، تنظيف الشاشة، VARIABLES, CLEAR SCREEN, *** - CURSOR HOME ارجع المشيرة.

تستخدم المتغيرات بكثرة في الحاسبات، وذلك لمحدودية استخدام الأرقام فيها، حيث تتطلب بعض البرامج عدداً كبيراً جداً من الأرقام ويمكن أن تكون المتغيرات أحرف أو أسماء تمثل أرقام معينة، تحدد المتغيرات أولا بأرقام ثابتة ثم تضاف أو تطرح منها مقادير معينة. لاحظ البرنامج (A1) التالي:

```
5   REM ***     A1     ****
10  REM *** VARIABLES ARE NUMBERS ***
20  CLS
30  A = 100: B = 200: C = 1.245
40  PRINT A; B; C
50  REM *** ADDING VARIABLES ***
60  REM *** TO MAKE NEW ONES ***
70  A1 = A + B
80  A2 = B * C
90  A3 = A/C
100 A4 = A + B - C
110 PRINT A1; A2; A3; A4
```

يمكن استخدام النقطتين (:) في سطر معين من البرنامج لفصل خطوتين عن بعضهما، وهذا ما نلاحظه في السطر 30 من البرنامج السابق. بهذه الطريقة يمكن وضع أكثر من خطوة واحدة في السطر الواحد. وسيظهر لك بلاغاً بالخطأ إذا لم تستخدم النقطتين في فصل الخطوات عن بعضها. البلاغ «CLS امسح» في السطر 20 من البرنامج السابق ينظف الشاشة تماماً ويضع «المشيرة CURSOR» في أعلى اليسار للشاشة. ويمكن لنا تحديد مكان المشيرة باستخدام البلاغ «LOCATE» وهذا سنأتي عليه لاحقاً. نفذ البرنامج الآن وستلاحظ أن الحاسب سيطبع التالي:

```
100  200   1.245
300  249   80.3212851405 62   298.755
OK
```

الآن نفذ البرنامج مرة أخرى وأثناء التنفيذ اضغط على المفتاح

«STOP قف» ستلاحظ توقف البرنامج، اضغط عليه مرة أخرى ليستمر البرنامج.

إذا ما استخدمت الفارزة المنقوطة (؛) مع الأمر «PRINT» نلاحظ أن الرموز طبعت إحداها بعد الآخر مباشرة، وليس بينها سوى فراغ بمقدار رمز واحد هو مكان علامة « + » التي تدل على أن الرمز يمثل رقم موجب، وإذا ما كان الرقم سالب فسنلاحظ وجود إشارة « - » أمامها.

*** TRING VARIABLES :*** المتغيرات الحزمية (الغير رقمية)

وهذه تمثل متغيرات تستخدم الأحرف والأسماء لتمثلها، بدلاً من الأرقام ويجب أن يكون آخر حرف فيها هو علامة الدولار «$». مثلا:

A$, NUM$, X2$

ويمكن لنا مساواة هذه المتغيرات إلى حزمة (مجموعة) من الأحرف أو الأسماء كما في المتغيرات الرقمية. مثلاً:

```
NAM$ = «COMPUTER»
```

نلاحظ هنا بأن مجموعة الأحرف يجب أن توضع بين فاصلتين علويتين (« »). وهذه المتغيرات يمكن أن تضاف إلى بعضها كما في حال المتغيرات الرقمية. مثلاً:

```
A$ = «JO»
B$ = «IN»
C$ = «ED UP»
D$ .= A$ + B$ + C$
```

وبذلك

```
D$ = «JOINED UP»
```

حاول الآن طباعة البرنامج (A2) التالي وانتبه إلى الفراغات في نهاية الكلمات، حيث يمثل جزء من الحزمة.

```
5     REM **** A2 ****
10    REM *** STRING VARIABLES ARE ***
20    REM *** NOT NUMBERS ***
30    REM *** STRINGS MUST BE SET ***
40    REM *** IN QUOTES ***
50    REM *** NOTE SPACE BEFORE WORDS ***
60    REM *** TO SEPARATE WHEN WORDS ***
70    REM *** ARE JOINED TOGETHER ***
80    REM
90    A$ = «M.S.X.»
100   B$ = «MICRO-COMPUTER»
110   C$ = «CAN»
120   D$ = «REMEMBER»
130   E$ = «WORDES»
140   F$ = «THE»
150   REM *** PRINT STRINGS TOGETHER ***
160   PRINT F$; A$; B$; C$; D$; E$
170   PRINT C$; F$; A$; B$; D$; E$
180   REM *** STRINGES CAN BE ADDED **
190   REM *** TO MAKE NEW ONES ***
200   AN$ = A$ + C$ + D$ + E$
210   PRINT AN$
```

ولكن لا يمكن مساواة المتغيرات الحـزمية إلى أرقام، والمتغيرات الرقمية إلى حزم من الأحرف، وفي حالة كهذه فإن بلاغاً بالخطأ سوف يظهر. مثلاً:

```
A$ = 10
```

أو

```
A = «BRIAN»
```

يكون خطأ.

أما

```
A$ = «100»
```

فهذا لا يمثل خطأ حيث إن الرقم «100» محصور بين حاصرتين علويتين وبذلك يمثل حزمة من الرموز وليس القيمة العددية للرقم «100». وهذا لا يمكن معالجته حسابياً. مثلاً:

```
A = 20
B = 30
C = A + B + 99
```

هذا صحيح حيث «C» سوف يساوي 149.

بينما

```
A$ = «20»
B = 30
C = A$ + B + 99
```

هذا خطأ.

إذا ما أضفنا حزماً من الرموز إلى بعضها فإنها سوف تطبع بعضها بعد بعض مباشرة. مثلاً:

```
A$ = «100»
B$ = «55»
C$ = A$ + B$
PRINT C$
```

وبذلك النتيجة تكون

10055

وهذا صحيح.

ويمكن للمتغيرات العددية وغير العددية (الحزمية) أن تكتب مع الأمر «PRINT» سويةً، وهذا ما سنلاحظه في البرامج التالية. انتبه إلى استخدام الفوارز (،) بدلاً من الفوارز المنقوطة (؛) في ذلك.

ففي السطر 90 من البرنامج (A3) التالي نحدد قيم المتغيرات العددية. والأسطر من 100 إلى 130 تعطي ما يقابل المتغيرات غير العددية من حزم.

```
9     REM ***      A3      ***
10    REM **** STRING VARIABLES AND
20    REM **** NUMERICAL VARIABLES MAY
30    REM **** BE USED IN THE SAME
40    REM **** PRINT STATEMENT
50    REM **** NOTE STARTING VARIABLES
60    REM **** AND NUMERICAL VARIABLES
```

```
70   REM **** CANNOT BE ADDED.  E.G.
80   REM **** A$ + D IS NOT ALLOWED
90   A = 20: B = 30: C = 40: D = 50
100  A$ = «TWENTY»
110  B$ = THIRTY»
120  C$ = «FORTY»
130  D$ = «FIFTY»
140  REM **** PRINT STRINGS AND
150  REM **** NUMERIC VARIABLES
160  PRINT A$, A
170  PRINT B$, B
180  PRINT C$, C
190  PRINT D$, D
200  REM *** COMMAS SET TO NEXT TAB
```

لاحظ بأننا استخدمنا الفارزة (،) بدلاً من الفارزة المنقوطة (؛) مع الأمر «PRINT»، وهذا يؤدي إلى زحف مناطق الطبع وكما لو كنا قد استخدمنا المفتاح «TAB» (أي بمقدار ثمانية فراغات).

*** حدد، عرض LOCATE, WIDTH ***

البلاغ «LOCATE» يستخدم لتحديد مكان المشيرة على الشاشة، وصيغة البلاغ تكون كالآتي:

```
LOCATE A, B
```

تمثل كل من «A» و«B» إحداثيات موقع المشيرة على الشاشة بالنسبة للمحورين «X» و«Y». والجهة العليا اليسرى من الشاشة لها الإحداثي (0,0) أي السطر «صفر»، والعمود «صفر» ولوضع المشيرة في الجهة العليا اليسرى يكون البلاغ بالصورة التالية:

```
LOCATE 0,0
```

وإذا ما أردنا أن نضع المشيرة في نهاية السطر، فإن ذلك يختلف حسب نوع الشاشة المستخدمة «0» أو «1» فإذا ما استخدمنا (الشاشة 0) فإن عرض الشاشة يكون (40) رمزاً، وإذا ما اتسخدمنا (الشاشة 1) فإن عرضها يكون (32) رمزاً فقط.

ويمكن لنا تحديد عرض غير ما ورد أعلاه للشاشة باستخدام البلاغ «عرض WIDTH»

```
5    REM *** A4 ***
10   REM *** WITH LOCATE COMMAND ***
20   A$ = «RED»: B$ = «BLUE»: C$ = «GREEN»
30   A = 1: B = 2: C = 3
40   REM *** CLEAR SCREEN ***
50   CLS
60   REM *** LOCATE AND PRINT NUMS ***
70   LOCATE 0,0
80   PRINT A
90   LOCATE 0,5
100  PRINT B
110  LOCATE 0,10
120  PRINT C
130  REM *** LOCATE AND PRINT WORD ***
140  LOCATE 10,0
150  PRINT A$
160  LOCATE 10, 5
170  PRINT B$
180  LOCATE 10,10
190  PRINT C$
```

في البرنامج (A4) السابق، يحدد السطر 20 قيم المتغيرات غير العددية ويحدد السطر 30 قيم المتغيرات العددية. أما السطر 50 فينظف الشاشة. والسطر 70 يطبع الرقم الأول. ونلاحظ بعد ذلك أن موقعاً جديداً يحدد للمشيرة في كل مرة لطباعة الأرقام والحزم. اطبع الآن السطر التالي ولاحظ الفرق.

WIDTH 29 (CR)

*** من ــ إلى ــ خطوة ــ تالي *** FOR-TO-STEP-NEXT

البلاغ «FOR... NEXT» هو دوار (حلقة)، وله أكثر من صورة، وسنأخذ أولاً أبسط هذه الصور، حيث يقوم الدوار بإجراء أو تنفيذ أوامر أو بلاغات لأكثر من مرة، ويوضع البلاغين «FOR» و«NEXT» في خطوتين منفصلتين في البرنامج، بحيث تكرر الخطوات بينهما عدة مرات، حسب ما مبين مع البلاغ «FOR». مثلاً:

```
FOR X = 1 TO 10
.....
.....
NEXT X
```

هذا يجعل البرنامج يكرر الخطوات المحصورة بينهما عشرة مرات.

حاول الآن طباعة البرنامج التالي:

```
5    REM **** A5 ****
10   REM *** EXAMPLE OF FOR-NEXT LOOP ***
20   FOR X = 1 TO 20
30   PRINT X
40   NEXT X
50   PRINT «THE LOOP HAS ENDED. X =»; X
60   END
```

نفذ البرنامج وستلاحظ النتيجة التالية:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
THE LOOP HAS ENDED. X = 21
```

ويمكن استخدام أي متغير في الدوار، بدلاً من المتغير «X»، حيث نضع في السطر 20 «X» مساوياً إلى «صفر»، ونطبع «X» في السطر 30، وثم نزيد من مقدار «X» في السطر 40 بمقدار واحد. وكذلك نفحص فيها إذا كانت «X» أكبر من «20» لتنهي الدوار.

ويمكن إضافة البلاغ «STEP خطوة» لتغيير مقدار الزيادة في قيمة المتغير. فإذا أردنا مثلاً أن تكون الزيادة بمقدار «2» بدلاً من «1» فإن صيغة البلاغ تصبح كالآتي:

```
FOR X = 1 TO 20 STEP 2
```

استخدم هذه الصيغة ولاحظ التغيير الذي يحدث في البرنامج السابق.

```
5     REM ****      A6      ****
10    REM *** EXAMPLE OF FOR-NEXT LOOP ***
20    FOR X = 1 TO 25
30    LOCATE X, 10: PRINT «*»
40    NEXT X
50    PRINT «THE LOOP HAS ENDED. X =»; X
60    END
```

في البرنامج السابق (A6) استخدمنا البلاغ «FOR..NEXT» للسيطرة على مكان الطبع على الشاشة، واستخدمنا المتغير «X» كذلك مع البلاغ «LOCATE»، الآن غير السطر 20 ليصبح كالآتي؛ ولاحظ الفرق.

```
20    FOR X = 10 TO 15
```

كذلك غير السطر 30 ليصبح:

```
30    LOCATE 10, X
```

ولاحظ النتيجة الآن.

اطبع البرنامج (A7) التالي، ولاحظ الفرق بينه وبين البرنامج السابق (A6) في استخدام البلاغ «FOR..NEXT».

```
5     REM ****        A7      ********
10    REM *** EXAMPLE OF FOR-NEXT LOOP ***
20    FOR X = .5 TO 5 STEP .5
30    PRINT X
40    NEXT X
50    PRINT «THE LOOP HAS ENDED. X =»; X
60    END
```

لاحظ أن الدوار يتوقف عندما تصبح قيمة «X» أكبر من «5» ولاحظ كذلك أن الزيادة «X» هي «0.5».

البرنامج (A8) التالي يستخدم البلاغ السابق أيضاً لكن مقدار «X» يبدأ بالنقصان. وفي هذه الحالة يجب أن نضع القيمة الكبرى لـ «X» أولاً ثم القيمة الصغرى. وهنا يجب استخدام البلاغ «STEP» ويكون مقداره سالباً.

```
5     REM ****      A8      ***
10    REM *** EXAMPLE OF FOR-NEXT LOOP ***
20    FOR X = 13 TO 3 STEP-1
30    PRINT X
40    NEXT X
50    PRINT «THE LOOP HAS ENDED. X =»; X
60    END
```

يمكن لنا استخدام المتغيرات بدلاً من الأرقام لتمثل قيم المتغير الأساسي المستخدم. لاحظ ذلك في البرنامج التالي (A9).

```
5     REM **** A9     ***
10    REM *** VARIABLES IN FOR-NEXT ***
20    REM
30    A = 100: B = 37: C = -1
40    A$ = «X   IS»
50    FOR X = A TO B STEP C
60    PRINT A$; X
70    NEXT X
80    PRINT «THE LOOP HAS ENDED. X =»; X
90    END
```

في البرنامج السابق، يثبت السطر 30 قيم المتغيرات ويستخدم السطر 50 البلاغ «FOR...NEXT».

ويمكن أن تكون المتغيرات المستخدمة مع البلاغ معادلة رياضية، وهذا ما نلاحظه في البرنامج (A10) التالي.

```
5     REM **** A10      ****
10    REM *** EQUATIONS IN FOR-NEXT ***
20    REM *** SCREEN STARTS ZERO ***
30    REM *** LOCATIONS ON SCREEN 0 ***
40    A$ = «X IS»
50    FOR X = 0 TO (40 * 24) - 1
60    PRINT A$; X
70    NEXT X
80    PRINT «THE LOOP HAS ENDED. X =»; X
90    END
```

أدناه بعض الأمثلة على استخدام البلاغ «FOR..NEXT» والتي يمكنك الاستفادة منها في كتابة برامجك.

```
FOR K = 0 TO 100
FOR V = 22 TO 77
FOR COLOUR = 0 TO 15
FOR CT = 100 TO 0 STEP  -2
FOR X = SCR TO SCR + 100
FOR AL = C - 2 TO B + C STEP B/10
FOR MM = 0.0034 TO 0.333 STEP 0.001
```

انتبه جيداً عند طباعتك «TO» في مكانها الصحيح واستخدم الحرف «O أو» وليس «0 الصفر» حيث طباعته بدلاً من حرف الـ «O» يعد من الأخطاء الشائعة. كذلك لاحظ الأرقام المستخدمة، فقد تخطىء بطباعة الحرف «O» بدلاً من الرقم «0».

*** ادخل *** INPUT

البلاغ «INPUT ادخل» يسمح لك بإدخال المعلومات إلى الحاسب في أثناء تنفيذ البرنامج فعندما يصل البرنامج إلى البلاغ «INPUT» فإن علامة استفهام « ؟ » تظهر، سائلة عن إدخال المعلومات المطلوبة. وتستخدم المتغيرات مع البلاغ، وتدخل الأرقام عند استخدام المتغيرات العددية والرموز والكلمات عند استخدام المتغيرات غير العددية. مثلاً:

INPUT A

يتطلب إدخال رقم

و

INPUT A$

يتطلب إدخال رمز أو حزمة من الرموز

وكما بيّنا سابقاً، فإن أية معلومات تطبع، لا تدخل الحاسب ما لم نضغط على مفتاح الإدخال (CR) وهذا ينطبق كذلك عند إدخال المعلومات مع البلاغ «INPUT» ويمكن كتابة عبارة في البرنامج تسبق الخطوة التي تحوي البلاغ، لتوضح المطلوب طباعته عند ظهور علامة الاستفهام « ؟ ». اطبع البرنامج (A 11) التالي ولاحظ ذلك.

```
5    REM **** A11     ***
10   REM *** INPUT VARIABLES FROM THE ***
20   REM *** KEYBOARD ***
30   REM *** NUMBERS ONLY ARE ALLOWED ***
40   REM
50   PRINT «TYPE IN A NUMBER AND PRESS CR KEY»
60   INPUT A
70   REM *** A = NUMBER ENTERED ***
80   PRINT «THE NUMBER YOU ENTERED WAS»; A
```

لاحظ أن السطر 50 يحتوي على عبارة توضح المعلومات المطلوب إدخالها، ويحتوي السطر 60 على البلاغ «INPUT». أما السطر 80 فإنه يطبع المعلومات والمتغير، وبذلك سنلاحظ أن الرقم الذي أدخلناه سوف يطبع على الشاشة. لنحاول الآن استخدام البلاغ «INPUT» لإدخال حزمة من الأحرف بدلاً من الأرقام.

```
5    REM **** A12     ****
10   REM *** INPUT  STRINGS FROM THE ***
20   REM *** KEY BOARD ***
30   REM *** ANY INPUT IS ALLOWED ***
40   REM
50   PRINT «TYPE IN ANY WORD AND PRESS CR KEY»
60   INPUT A$
70   REM *** A$ = STRING ENTERD ***
80   PRINT «THE STRING YOU ENETERD IS:»; A$
```

نلاحظ في البرنامج (A12) السابق أن السطر 50 يطبع العبارة التوضيحية للمعلومات المطلوبة، والسطر 60 يحتوي على البلاغ «INPUT»، أما السطر 80 فإنه يطبع العبارة التوضيحية والاسم الذي أدخل للبرنامج أثناء التنفيذ.

البرنامج (A13) التالي يستخدم الدوار «FOR..NEXT» لإعادة استخدام البلاغ «INPUT» لأكثر من مرة.

```
5     REM *** A13     ***
10    REM *** INPUT AND FOR-NEXT ***
20    REM *** FOR MULTIPLE  ENTRIES  ***
30    REM
40    REM *** INPUT MESSAGE IN QUOTES ***
50    REM
60    FOR I = 1 TO 5
70    INPUT «ENTER NAME AND PRESS CR KEY»; A$
80    INPUT «ENTER TELEPHONE NUMBER AND CR»; A
90    PRINT A$; A
100   NEXT I
```

لاحظ السطرين 70, 80 ولاحظ أن المعلومات قد طبعت في هذين السطرين محصورة بين حاصرتين علويتين (« ») وهذا يعوض عن وضع خطوة أخرى تسبقهما، وتحتوي على المعلومات، كما في البرامج السابقة. انتبه جيداً إلى وجود الفارزة المنقوطة في السطر حيث أن موقعها ضروري، ويجب أن لا تنساها أبداً. يستخدم السطر 60 الدوار «FOR...NEXT» ويطلب السطر 70 إدخال حزمة من الأحرف. أما السطر 80 فيطلب إدخال رقم. وثم يطبع السطر 90 الحزمة والرقم سوية.

نستخدم في البرنامج (A14) القادم الفكرة نفسها للبرنامج السابق للحصول على حاصل جمع ثلاثة أعداد ندخلها إلى الحاسب في أثناء تنفيذ البرنامج.

```
5     REM **** A14
10    REM **** INPUT AND FOR NEXT
20    REM **** FOR MULTIPLE ENTRIES
30    REM ****
40    REM **** ADDING THREE NUMBERS
```

```
60    FOR N = 1 TO 3
70    INPUT «ENTER A NUMBER AND PRESS CR KEY»; A
80    B = B + A
90    NEXT N
100   PRINT «THREE NUMBERS WHERE ENTERED»
110   PRINT «AND ADDED TOGETHER THEY EQUAL»; B
```

لاحظ السطر 70 في البرنامج السابق، حيث استخدم البلاغ «INPUT»، وفي السطر 80 أضيفت الأعداد إلى بعضها البعض. ويستمر البرنامج لثلاث دورات لإدخال ثلاثة أعداد.

أعد تنفيذ البرنامج عدد من المرات وفي كل مرة استخدم أعداداً مختلفة ولاحظ النتائج.

*** كشف $ *** INKEY$

يعتبر البلاغ «INKEY$» طريقة أخرى لإدخال المعلومات إلى الحاسب في أثناء تنفيذ البرنامج، وباستخدام لوحة المفاتيح. وهـذا يختلف عن البلاغ «INPUT»، حيث لا ينتظر منا الحاسب إدخال المعلومات المطلوبة بل يفحص فيما إذا كنا قد طبعنا رمزاً معيناً أم لا، ففي حالة طباعته يذهب الحاسب إلى سطر معين في البرنامج. وفي حالة عدم طبـاعته يـذهب إلى سطر آخـر. والمعلومات التي تدخل إلى الحاسب تكون رمزاً (حرفاً) واحداً فقط. ويستخدم البلاغ بالصورة التالية:

```
Q$ = INKEY$
```

لاحظ ذلك في البرنامج (A15) التالي، حيث يطبع الحاسب أي رمز تدخله من لوحة المفاتيح.

```
5     REM **** A15 ***
10    REM *** INKEY$ COMMAND EXAMPLE ***
20    REM *** IF NO KEY IS PRESSED THE ***
30    REM *** STRING IS CLEARED ***
40    FOR X = 1 TO 1000
50    A$ = INKEY$
60    PRINT A$;
70    NEXT X
```

لاحظ أن البلاغ «PRINT» في السطر 60 تُبعَ بفارزة منقوطة (؛) وهذا يؤدي إلى أن يكون الطبع للرمز التالي مجاوراً للرمز الذي يسبقه، وليس تحته. ويمكن كذلك استخدام الفارزة (،) بدلاً من الفارزة المنقوطة (؛) وبذلك تتباعد الرموز بمسافة ثمانية فراغات عن بعضها البعض (كما لو كنا قد استخدمنا الدالة «TAB»). أعد تنفيذ البرنامج السابق مرةً أخرى بعد إبدال الفارزة المنقوطة بفارزة (،) ولاحظ الفرق.

*** VAL *** قيمة

لاحظنا أن الأرقام يمكن إدخالها ومعالجتها في الحاسب كرموز وليس كقيم عددية. وفي بعض الأحيان يتطلب الأمر معالجتها بقيمتها العددية في الوقت نفسه، لهذا تستخدم الدالة «VAL قيمة» للحصول على القيمة العددية لرمز معين. والصيغة لذلك تكون:

```
A$ = «7»
A = VAL (A$)
```

هذا يؤدي إلى الحصول على قيمة الرمز «7» مع الاحتفاظ بالرمز «A$» مساوياً للرمز «7». وأن قيمة أي رمز آخر غير الأرقام يكون صفراً. لاحظ الأمثلة التالية:

```
A$ = «1.13»
A = VAL (A$)
A = 1.13

A$ = «X40»
A = VAL(A$)
A = 0

A$ = «40X»
A = VAL(A$)
A = 40

A$ = «QWERTY»
A = VAL(A$)
A = 0
```

البرنامج (A16) التالي يستخدم البلاغ «INKEY$» لإدخال الرموز من لوحة المفاتيح ويحولها إلى قيمتها العددية. لاحظ أن القيمة العددية للأرقام تكون مساوية لها وللرموز الأخرى تساوي «صفر».

```
5    REM **** A16     ****
10   REM *** INKEY$ AND VAL COMMANDS ***
20   REM *** IF NO KEY IS PRESSED THE ***
30   REM *** ANSWER IS ZERO ***
40   FOR X = 1 TO 10
50   A$ = INKEY$
60   PRINT VAL(A$)
70   NEXT X
```

لاحظ أن البلاغ «PRINT» في السطر 60 تبع بفرزة منقوطة « ؛ »، وهذا يؤدي إلى طباعة النتائج بصورة متجاورة. أبدل الفارزة المنقوطة « ؛ » بفارزة « ، » ولاحظ الفرق. كذلك لاحظ أن البلاغ «PRINT» والدالة «VAL» وضعا في خطوة واحدة.

*** أدخل $ *** INPUT $

الدالة «INPUT$» تعمل عمل البلاغين «INPUT» و «INKEY$» سويةً. وتستخدم بالصيغة التالية:

```
INPUT$(3)
```

الرقم (3) أعلاه يعني إدخال ثلاثة رموز، ويمكن استخدام أي رقم لتحديد عدد الرموز التي يجب إدخالها. تظهر مع هذه الدالة علامة الاستفهام « ؟ » على الشاشة، ولا تطبع أية جملة، كما في البلاغ «INPUT».

بعد إدخال الرموز كافة يستمر البرنامج، فإذا ما كانت الرموز التي أدخلت أقل من الرموز التي تم تحديدها في الدالة، اضغط على المفتاحين «CTRL + C» سوية لاستمرار البرنامج.

```
5     REM **** A17      ****
10    REM *** INPUT$ COMMANDS ***
20    REM *** NUMBER IN BRACKETS IS ***
30    REM *** NUMBER OF CHARACTERS ***
40    CLS
50    FOR X = 1 TO 10
60    A$ = INPUT$(5)
70    PRINT A$
80    NEXT X
```

البرنامج (A17) السابق يطلب منك إدخال خمسة رموز ويضعها بعد ذلك مساوية لـ «A$». ثم يطلب البرنامج من الحاسب طباعة «A$»، ويستمر البرنامج لعدة دورات، وكما حدد له ذلك في الدوار «FOR..NEXT» في السطر 40. أدخل أقل من خمسة رموز، ثم اضغط على المفتاحين «CTRL + C» سوية لتجعل الحاسب يستمر في البرنامج.

اطبع البرنامج (A18) التالي، ولاحظ أن عدد الرموز التي حددها البلاغ ما هي إلا متغير، يتغير تباعاً للدوار «FOR..NEXT» في السطر 40 من البرنامج.

```
5     REM **** A18 ****
10    REM *** INPUT$ COMMANDS ***
20    REM *** VARIABLE IN BRACKETS IS ***
30    REM *** NUMBER OF CHARACTERS ***
40    FOR X = 1 TO 10
50    IP$ = INPUT$(X)
60    PRINT IP$
70    NEXT X
```

*** DIM, ARRAY

*** بُعْد، نسق (مصفوفة)

المصفوفات ما هي إلا مجموعة من المتغيرات، يحدد عددها باستخدام البلاغ «DIM بعد». ويحدد اسم ومقدار المصفوفة باستخدام اسم ورقم محصور بين قوسين بعد البلاغ «DIM». مثلاً:

```
B(6)
```

اسم المصفوفة هنا هو «B»، ومقدارها «6»، أي لها سبعة مقاطع تبدأ بـ «الصفر» وتنتهي بالرقم «ستة»، وكالآتي:

B(0)
B(1)
B(2)
B(3)
B(4)
B(5)

هذه كلها تمثل متغيرات منفصلة تكون المصفوفة. ويمكن معالجة أجزاء المصفوفة منفصلة كما في معالجة المصفوفة الكاملة. وكما يلي:

B(0) = 100

أو B(4) = 22.734

أو B(2) = X

أو PRINT B(3)

أو X = B(A)

وهكذا.

يستخدم البرنامج (A19) التالي الدوار «FOR..NEXT» لتغيير مقدار «X» بزيادته بصورة مستمرة. وحددت قيمة «X» هنا من «صفر» إلى «5». ويسمح لك السطر 80 بإدخال الأرقام داخل المصفوفة من «A(0)» إلى «A(5)».

```
5   REM **** A19 *****
10  REM *** SINGLE ARRAYS ***
20  REM *** DIMENSION ARRAYS FIRST ***
30  REM
40  DIM A(5)
50  REM *** FILL ARRAYS ***
60  FOR X = 0 TO 5
70  PRINT «ENTER NUMBER AND PRESS CR»
80  INPUT A(X)
90  NEXT X
100 REM *** DISPLAY ARRAY CONTENTS ***
110 REM ***  IN REVERSE ***
120 FOR X = 5 TO 0 STEP -1
130 PRINT A(X)
140 NEXT X
```

يمكن استخدام الرموز في المصفوفة بدلاً من الأرقام. وتنطبق كل شروط استخدام الرموز غير العددية على ذلك ويجب تحديد أبعاد المصفوفة التي عناصرها رموز وليست أعداد قبل استخدامها. مثلاً:

```
DIM A$(5)
```

هذ يعني أن المصفوفة مكونة من:

```
A$(0)
A$(1)
A$(2)
A$(3)
A$(4)
A$(5)
```

كل واحد من هذه تمثل متغيراً واحداً غير عددي وبمجموعها تمثل المصفوفة غير العددية (الرمزية). ويمكن كذلك إعطاء اسم للمصفوفة الرمزية بنفس طريقة إعطاء الأسماء للمتغيرات الرمزية، وكذلك يمكن معالجة أجزاء المصفوفة الرمزية كما في المتغيرات الرمزية. الأمثلة التالية توضح ذلك:

```
B$(0) = «1012»
B$(5) = «12.3314»
B$(2) = A$
PRINT B$(3)
Z$ = B$(A)
```

البرنامج (A20) التالي يستخدم المصفوفات الرمزية بدلاً من المصفوفات العددية. اطبع البرنامج ولاحظ ذلك.

```
5     REM **** A20 ****
10    REM *** STRING ARRAYS ***
20    REM *** DIMENSION ARRAYS FIRST ***
30    REM
40    DIM  A$(5)
50    REM *** FILL STRING ARRAYS ***
60    FOR Z = 0 TO 5
70    PRINT «ENTER WORD AND PRESS CR»
80    INPUT A$(Z)
90    NEXT Z
100   REM *** DISPLAY ARRAY CONTENTS ***
110   REM *** IN REVERSE ***
```

```
120   FOR Z = 5 TO 0 STEP -1
130   PRINT A$(Z)
140   NEXT Z
```

*** READ... DATA

*** اقرأ. . . بيانات

يستخدم البلاغين «READ اقرأ» و«DATA بيانات» سوية في البرنامج لحفظ عدد من الأرقام أو البيانات وقراءتها عند الحاجة لها، لاستخدامها في البرنامج. وتستخدم المتغيرات لمرة واحدة في البرنامج فقط، فإذا ما أردنا استخدامها لأكثر من مرة، يجب وضعها في خطوة خاصة بها. والبلاغ «READ» يقرأ البيانات ويحفظها ضمن المتغيرات أو المصفوفات الموجودة في البرنامج ليرجع إليها عند الحاجة.

هناك نوعان من البيانات يمكن استخدامها مع البلاغ «DATA»، البيانات العددية والبيانات الرمزية (غير العددية). وتكون صيغة البلاغ «DATA» كالآتي:

```
DATA COLOUR, SHIPE, SIZE 1, 2, 3,
```

حيث يفصل كل جزيء من أجزاء البيانات عن الجزء الآخر باستخدام الفارزة وتكون صيغة البلاغ «READ» بالشكل التالي:

```
READ A
READ A$
READ A$(3)
```

لاحظ في المثال السابق أن الأجزاء الثلاثة الأولى من البيانات هي حزم من الرموز (ليس بأعداد) والأجزاء الثلاثة الأخيرة أعداد يجب أن تقرأ هذه بصورة صحيحة، حيث يجب استخدام المتغيرات العددية مع الأعداد، والمتغيرات الرمزية مع الرموز، وإلا سيعطي الحاسب إشارة بالخطأ.

عند تنفيذ البرنامج يقوم البلاغ «READ» بقراءة الجزء الأول من البيانات ثم الجزء الثاني، وثم الثالث، وهكذا. . .

ويعطي الحاسب إشارة بالخطأ إذا ما استخدمت البلاغ «READ» دون إعطاء بيانات في البرنامج، أو إذا ما استخدمت البلاغ عدداً من المرات يزيد على عدد البيانات المعطاة. نفذ البرنامج (A21) التالي، ولاحظ ذلك.

```
5     REM *** A21 ***
10    REM *** READ DATA STATEMENT ***
20    REM *** DATA MUST MATCH ***
30    REM *** VARIABLES TYPE.  I.E. ***
40    REM *** STRING DATA INTO ***
50    REM *** STRING VARIABLE ***
60    REM *** NUMERICAL DATA INTO ***
70    REM *** NUMERIC VARIABLES ***
80    REM
90    DIM A$(5): DIM A(5)
100   FOR X = 0 TO 5
110   READ A$(X): READ A(X)
120   NEXT X
130   REM *** DISPLAY ARRAYS ***
140   PRINT «NAME», «NUMBER»
150   FOR X = 0 TO 5
160   PRINT A$(X), A(X)
170   NEXT X
180   DATA ALI, 254752, BASIM, 739277, SALIH, 435724
190   DATA RIYADH, 864663, HAYDAR, 64397, MAHMOOD, 963954
```

البيانات في السطرين 180 و 190 مستمرة وبذلك فإن البيانات التي لدينا مكونة من اثني عشر جزءاً، منها بيانات عددية ومنها رمزية. يقرأ بلاغاً «READ» في السطر 110 البيانات بالتسلسل، حيث يقرأ الأول البيانات الرمزية، والثاني البيانات العددية. ويحدد الدوار في السطر 150 قراءة البيانات ولستة مرات، وبذلك تُحَمَل البيانات كمصفوفة، وكالآتي:

A$(0) ALI	A(0) 254752
A$(1) BASIM	A(1) 739277
A$(2) SALIH	A(2) 435724
A$(3) RIYADH	A(3) 864663
A$(4) HAYDAR	A(4) 64397
A$(5) MAHMOOD	A(5) 963954

توضع الأرقام بين حاصرتين علويتين إذا ما استخدمت كـرموز وليس كأعداد. مثلاً:

DATA TEN, TWENTY, «2000», «A,B»

يعامل الرمزين (A) و (B) كعنصر واحد في المثال السابق رغم وجود الفارزة بينهما، وذلك لحصرهما سوية بين حاصرتين علويتين، وبذلك يكون في المثال السابق أربعة عناصر وليس خمسة.

*** RESTORE

*** عاود

تعلمنا مما سبق أن البلاغ «READ» يقرأ الجزء الأول من البيانات، ثم الثاني، وثم الثالث، وهكذا. . . ، وإذا ما أردنا الرجوع إلى البيانات نفسها مرة أخرى، يجب استخدام البلاغ «RESTORE». يستخدم البرنامج (A22) التالي خمسة عناصر من البيانات في السطر 270 ولدينا 20 متغيراً يجب ملؤها بهذه البيانات، لذلك استخدمنا البلاغ «READ» في السطر 60 لقراءة البيانات التي حدد عددها بواسطة الدوار «FOR..NEXT» في السطر 50 بخمسة بيانات. ولإعادة قراءتها مرة أخرى بواسطة البلاغ «READ» في السطر 110 يجب أن نسبقه بالبلاغ «RESTORE»، وإلا ستظهر إشارة بالخطأ مشيرة إلى أن البيانات غير كافية. البرنامج (A22) التالي يستخدم البلاغ «RESTORE» في السطر 80 لإعادة قراءة البيانات لأكثر من مرة.

```
5   REM **** A22     ***
10  REM *** RESTORE WILL RESET DATA ***
20  REM *** TO USE AGAIN ***
30  REM
40  DIM A$(19)
50  FOR X = 0 TO 4
60  READ A$(X)
70  NEXT X
80  RESTORE
90  REM *** READ ERROR IF NO RESTORE ***
100 FOR X = 5 TO 9
110 READ A$(X)
120 NEXT X
```

```
130   RESTORE
140   REM *** READ ERROR IF NO RESTORE ***
150   FOR X = 10 TO 14
160   READ A$(X)
170   NEXT X
180   RESTORE
190   REM *** READ ERROR IF NO RESTORE ***
200   FOR X = 15 TO 19
210   READ A$(X)
220   NEXT X
230   REM *** DISPLAY ARRAY ***
240   FOR X = 0 TO 19
250   PRINT A$(X)
260   NEXT X
270   DATA ALI      RIYADH, HAYDAR HASAN, MAHDI AHMAD,
      SAWSAN KHALID, AMAL   YOUSIF
```

*** GOTO *** اقصد

البلاد «GOTO اقصد»، بلاغ بسيط للغاية، ولكن له فائدة كبيرة، حيث يؤدي استخدامه إلى انتقال الحاسب من السطر الذي فيه البلاغ إلى السطر المحدد. مثلاً:

```
20    GOTO 100
```

هذا يقود الحاسب إلى القفز من السطر 20 إلى السطر 100 دون المرور بالخطوات التي بينهما.

لاحظ البرنامج (A23) القادم حيث استخدم البلاغ «GOTO» في السطر 70 لإرجاع الحاسب إلى السطر 50، وبذلك يقوم بتنفيذ السطرين 50 و 60.

```
5     REM **** A23     ****
10    REM *** GOTO COMMAND ***
20    REM *** NOTE  PRINT SEQUENCE ***
30    REM *** GOTO CHANGES ORDER ***
40    REM
50    PRINT «FIRST PRINT STATEMENT»
60    PRINT «SECOND PRINT STATEMENT»
70    GOTO 110
```

```
80    PRINT «THIRD PRINT STATEMENT»
90    PRINT «FOURTH PRINT STATEMENT»
100   GOTO 170
110   PRINT «FIFTH PRINT STATEMENT»
120   GOTO 150
130   PRINT «SIXTH PRINT STATEMENT»
140   GOTO 80
150   PRINT «SEVENTH PRINT STATEMENT»
160   GOTO 130
170   END
```

ويستخدم البلاغ أيضاً لتوجيه الحاسب لتنفيذ البرنامج متسلسلاً بتسلسل غير التسلسل الذي كتب به البرنامج. نفذ البرنامج (A24) التالي، ولاحظ تسلسل طباعة الجمل.

```
5     REM **** A24      *****
10    REM *** GOTO COMMAND ***
20    REM *** NOTE PRINT SEQUENCE ***
30    REM *** GOTO CHANGES ORDER ***
40    REM
50    PRINT «FIRST PRINT STATEMENT»
60    PRINT «SECOND PRINT STATEMENT»
70    GOTO 110
80    PRINT «THIRD PRINT STATEMENT»
90    PRINT «FOURTH PRINT STATEMENT»
100   GOTO 170
110   PRINT «FIFTH PRINT STATEMENT»
120   GOTO 150
130   PRINT «SIXTH PRINT STATEMENT»
140   GOTO 80
160   GOTO 130
170   END
```

*** تفرع . . . ارجع *** GOSUB... RETURN

لا يختلف البلاغ «GOSUB تفرع» عن البلاغ «GOTO» إلا قليلاً. فعندما يصل البرنامج إلى البلاغ «GOSUB» ينتقل الحاسب إلى السطر المبين فيه كما في البلاغ «GOTO» ولكن الاختلاف هنا هو أن الحاسب يتذكر من أي سطر

قد تفرع البرنامج ليعود إليه حال وصوله إلى البلاغ «RETURN»، ويبدأ من السطر الذي يلي السطر الذي بدأ منه.

لاحظ البرنامج (A25) التالي جيداً، فهو يبدأ بالسطر 50 حيث يطلب الحاسب إدخال ثلاثة أرقام، ثم ينتقل إلى السطر 160 وينفذ الخطوات الواردة في الأسطر 190,180,170. ثم يعود إلى السطر 70 ليطلب ثلاثة أرقام أخرى وينتقل إلى السطر 160 مرة أخرى، وينفذ الأسطر 190,180,170 ويرجع ثانية إلى السطر 90 ويستمر حتى السطر 190 ويتوقف لوجود البلاغ «END». استخدام البلاغ «END» هنا ضروري، وإلا فإن الحاسب يستمر ويدخل في البرنامج الفرعي الذي يبدأ في السطر 160، ويعطي بذلك إشارة بالخطأ، وتكتب البرامج الفرعية عادة في نهاية البرنامج الرئيسي ويمكن استخدام أكثر من برنامج فرعي واحد ضمن البرنامج الرئيسي.

```
5     REM **** A25     ***
10    REM **** GOSUB STATEMENT
20    REM **** GOSUB CALL MUCH USED
30    REM **** ROUTINE TO SAVE MEMORY
40    REM ****
50    INPUT «ENTER 3 NUMBERS»; A, B, C
60    GOSUB  160
70    INPUT «ENTER ANOTHER 3 NUMBERS»; A, B, C
80    GOSUB  160
90    PRINT «THIS IS ALL»
100   REM **** THE END STATEMENT IS USED
110   REM **** TO STOP THE PROGRAM
120   REM **** RUNNING INTO THE ROUTINE
130   REM **** AND CAUSING A RETURN ERR.
140   END
150   REM **** SUBROUTINE WITH RETURN
160   ANS = A + B + C
170   PRINT «THE COMPUTER HAS ADDED THE»
180   PRINT «THREE NUMBERS YOU ENTERED»
190   PRINT «AND THE RESULT IS»; ANS
200   RETURN
```

*** ON...GOTO, ON...GOSUB

*** نعم.. اقصد، نعم.. تفرع

يستخدم الإيعاز «ON» مع البلاغين «GOTO» و«GOSUB» وذلك لإعطاء استخدام أفضل للإعازين. وتكون الصيغة كالآتي:

```
ON X GOTO 100, 200, 300
ON X GOSUB 100, 200, 300
```

استخدمنا المتغير «X» في المثالين السابقين، ويمكن استخدام أي متغير آخر، وتمثل أرقام الأسطر بصورة متسلسلة 1، 2، 3، وهكذا فإذا ساوى المتغير لأحد هذه الأرقام فإن البرنامج يقصد أو يتفرع إلى السطر الذي يناظر هذا الرقم. فإذا ما ساوى المتغير «X» إلى الرقم «2» في المثال السابق، فإن الحاسب يقصد أو يتفرع إلى السطر 200. أما إذا ساوى إلى رقم أكبر من الأرقام المبينة، فإن الحاسب سيستمر بالبرنامج دون أن يقصد سطر آخر أو يتفرع إليه.

أدناه ثلاثة برامج (A28, A27, A26) نستخدم فيها البلاغان «ON.. GOSUB, ON.. GOTO» نفذ هذه البرامج ولاحظ تأثير البلاغين.

```
5   REM **** A26     *****
10  REM *** ON-GOTO STATEMENT ***
20  REM *** GOTO LINE DEPENDENT ON ***
30  REM *** VARIABLE ***
40  REM
50  INPUT «ENTER A NUMBER FROM 1 TO 4»; A
55  PRINT
60  ON A GOTO 90, 110, 130, 150
70  PRINT «THAT WAS NOT A VALID NUMBER»
75  PRINT
80  GOTO 50
90  PRINT «THE KEY YOU PRESSED WAS ONE»
95  PRINT
100 GOTO 50
110 PRINT «THE KEY YOU PRESSED WAS TWO»
115 PRINT
120 GOTO 50
125 PRINT
130 PRINT «THE KEY YOU PRESSED WAS THREE»
135 PRINT
```

```
140   GOTO 50
150   PRINT «THE KEY YOU PRESSED WAS FOUR»
155   PRINT
160   GOTO 50
```

```
5     REM **** A27      ******
10    REM *** ON-GOSUB STATEMENT ***
20    REM *** GOSUB  LINE DEPENDENT ON ***
30    REM *** VARIABLE ***
40    REM
50    INPUT «ENTER A NUMBER FROM 1 TO 4»; A
60    ON A GOSUB 160, 220, 280, 340
70    GOTO 50
80    END
130   REM *** THE  END COMMAND WILL ***
140   REM *** STOP THE PROGRAM RUNNING ***
150   REM *** INTO THE SUBROUTINES ***
160   REM *** SUBROUTINE ONE ***
170   FOR X = 1 TO 10
180   PRINT «THIS IS THE FIRST SUBROUTINE»
190   NEXT X
200   PRINT «WE WILL NOW RETURN TO THE MAIN PROGRAM»
210   RETURN
220   REM *** SUBROUTINE TWO ***
230   FOR X = 1 TO 10
240   PRINT «THIS IS THE SECOND SUBROUTINE»
250   NEXT X
260   PRINT «WE WILL NOW RETURN TO THE MAIN PROGRAM»
270   RETURN
280   REM *** SUBROUTINE ***
290   FOR X = 1 TO 10
300   PRINT «THIS IS THE THIRD SUBROUTINE»
310   NEXT X
320   PRINT «WE WILL NOW RETURN TO THE MAIN PROGRAM
330   RETURN
340   REM *** SUBROUTINE FOUR ***
350   FOR X = 1 TO 10
360   PRINT «THIS IS THE FOURTH SUBROTINE»
370   NEXT X
380   PRINT «WE WILL NOW RETURN TO THE MAIN PROGRAM»
390   RETURN
```

```
5   REM **** A28      *****
10  REM *** ON-GOSUB  STATEMENT ***
20  REM *** TO GET NAME FOR NUMBER ***
40  PRINT «ENTER NUMBER AGAINST NAME»
50  PRINT «ENTER NUMBER AGAINST NAME»
60  PRINT
70  PRINT «1. ALI»
80  PRINT «2 . AHMAD»
90  PRINT «3 . HAYDAR»
100 . PRINT «4 . RIYADH»
110 INPUT A
120 ON A GOSUB 170, 190, 210, 230
130 PRINT «TRY AGAIN»
140 GOTO 50
150 REM *** GOTO STOPS PROGRAM ***
160 REM *** RUNNING THROUGH ROUTINES ***
170 PRINT «ALI TELEPHONE NUMBER IS 3456644»
180 RETURN
190 PRINT «AHMAD TELEPHONE NUMBER IS 2567543»
200 RETURN
210 PRINT «HAYDAR TELEPHONE NUMBER IS 8568444»
220 RETURN
230 PRINT «RIYADH TELEPHONE NUMBER IS 1937364»
240 RETURN
```

*** إذا . . . إذن

*** IF... THEN

يستخدم البلاغ «IF...THEN إذا . . . إذن» لتحديد عمل الحاسب اعتماداً على حدث معين. ويستخدم البلاغ بالصيغة التالية:

IF (this is true) **THEN** (do this)

إذا (كان هذا صحيحاً) إذن (أجرِ التالي)

يفحص البرنامج الجملة الموجودة بعد «IF» فإذا كانت صحيحة، ينفذ الخطوة بعد «THEN» مباشرة. وإذا لم تكن صحيحة فإن الحاسب يهمل الخطوة بعد «THEN»، ويستمر. و بعض الأمثلة على ذلك:

```
10 = 10
X = 3
G = SAR
```

```
A$ = «BLUE»
G > 5
F > H
SCR < > SI
```

يمكن وضع أي إيعاز خاص بلغة البيسك بعد الإيعاز «THEN». مثلاً:

```
IF read$ = «william» THEN print «this is correct»
```

هذا يعني عندما يكون المتغير «read$» مساوياً لـ «william» فقط، يطبع الحاسب العبارة «this is correct».

البرنامج التالي يستخدم فيه البلاغ «IF...THEN» حيث يفحص الحاسب المتغير المعطى له ليعطي الجواب.

```
5     REM **** A29     ***
10    REM *** IF-THEN STATEMENT ***
20    REM *** IF STATEMENT IS TRUE ***
30    REM *** THEN ACTION  ***
40    REM *** IF NOT THEN DO NOTHING ***
60    PRINT «ENTER YES OR NO»
70    INPUT A$
80    IF A$ = «YES» THEN GOTO 110
90    PRINT «YOU ENTERED NO»
100   GOTO 60
110   PRINT «YOU ENTERED YES»
120   GOTO 60
```

البرنامج السابق يسأل عن كلمة «YES» فإن لم تكن الإجابة «YES» يفترض أن تكون «NO». ويفسّر الحاسب أي كلمة غير كلمة «YES» أنها «NO». إذا ما احتجنا بأن نجعل الحاسب يعالج المتغيرات التي نطلبها فقط ويهمل الأخرى، يجب أن يفحص البرنامج كافة المتغيرات، وهذا ما نلاحظه في البرنامج (A30) التالي:

```
5     REM **** A30     ****
10    REM *** IF-THEN STATEMENT ***
20    REM *** IF STATEMENT IS TRUE ***
30    REM *** THEN ACTION  ***
40    REM *** IF NOT THEN DO NOTHING ***
```

```
60    PRINT «ENTER NUMBER 1 TO 4»
70    INPUT A
80    IF A = 1 THEN GOTO 200
90    IF A = 2 THEN GOTO 300
100   IF A = 3 THEN 400
110   IF A = 4 THEN 500
120   REM *** NOTE GOTO MAY BE OMMITED ***
130   REM *** AS IN LINES 100 AND 110 ***
140   GOTO 70
200   FOR X = 1 TO 50
210   PRINT «ONE»
220   NEXT X
230   GOTO 60
300   FOR X = 1 TO 50
310   PRINT «TWO»
320   NEXT X
330   GOTO 60
400   FOR X = 1 TO 50
410   PRINT «THREE»
420   NEXT X
430   GOTO 60
500   FOR X = 1 TO 50
510   PRINT «FOUR»
520   NEXT X
530   GOTO 60
```

يمكن الاستغناء عن البلاغ «GOTO» بعد البلاغ «THEN» ووضع رقمٍ السطر فقط. أما البلاغات الأخرى غير البلاغ «GOTO» يجب أن تطبع كاملةً

البرنامج (A31) التالي مثال آخر على استخدام البلاغ «IF...THEN»

```
5     REM **** A31     ****
10    REM *** IF THEN STATEMENT ***
20    REM *** IF STATEMENT IS TRUE ***
30    REM *** THEN ACTION ***
40    REM *** IF NOT THEN DO NOTHING ***
60    PRINT «ENTER NUMBERS 1 TO 4»
70    INPUT A
80    IF A = 1 THEN PRINT «ONE»
90    IF A = 2 THEN PRINT «TWO»
100    IF A = 3 THEN PRINT «THREE»
```

```
110   IF A = 4 THEN PRINT «FOUR»
120   GOTO 60
```

*** ELSE *** وإلا

يستخدم الإيعاز «ELSE وإلا» مع البلاغ «IF...THEN» لإعطاء النتيجة المطلوبة إذا لم تكن العبارة بعد «IF» صحيحة. والصيغة تكون بالشكل التالي:

IF (this is true) **THEN** (do this) **ELSE** (do that)

إذا (كان هذا صحيحاً) إذن (أجرِ الأمر التالي) وإلا (أجرِ الأمر الآخر)

حيث يفحص البرنامج العبارة بعد «IF» فإذا ما وجدها صحيحة ينفذ الإيعاز بعد «THEN» وإذا لم تكن صحيحة فإنه ينفذ الإيعاز الذي بعد «ELSE». ويمكن استخدام أي إيعاز بعد الدالة «ELSE» من الإيعازات الخاصة بلغة البيسك. لاحظ المثال التالي:

IF SP = 1045 THEN print «this answer is 1045» ELSE print «no it is not»

هذا يعني أن الحاسب سوف يطبع العبارة الأولى عندما يساوي المتغير «SP» إلى «1045» فقط وإلا فإنه يطبع العبارة الثانية.

البرنامج (A32) التالي يستخدم الإيعاز السابق لفحص المتغير الذي يعطى إلى الحاسب، ويطبع الجواب المناسب له.

```
5     REM **** A32     ***
10    REM *** IF-THEN-ELSE STATEMENT  ***
20    REM *** IF STATEMENT IS TRUE ***
30    REM *** THEN ACTION     ***
40    REM *** ELSE ANOTHER     ***
50    PRINT «ENTER BLACK OR WHITE»
60    A$ = INPUT$ (5)
70    IF A$ = «BLACK» THEN PRINT «I MOVE»; ELSE PRINT «YOU
      MOVE»;
80    PRINT «FIRST»
90    GOTO 50
```

السطر 60 في البرنامج السابق يسأل عن أحد الاحتمالين، هذا يعني أن الحاسب سوف يفحص واحداً فقط وهنا هو «BLACK» فإذا لم يكن «BLACK» يجب أن يكون «WHITE». وبطبيعة الحال إذا ما طبعنا أي كلمة أخرى غير كلمة «BLACK» يعتبرها الحاسب «WHITE».

إذا ما أردنا من الحاسب أن يهمل أي عبارة تدخل إليه عدا العبارات التي حددت له مسبقاً، يجب أن يقوم الحاسب بفحص جميع العبارات نسبةً إلى العبارات التي حددت له مسبقاً. البرنامج (A33) أدناه يوضح ذلك.

```
5     REM **** A33     ***
10    REM *** IF-THEN-ELSE STATEMENT ***
20    REM *** IF STATEMENT IS TRUE ***
30    REM *** THEN ACTION     ***
40    REM *** ELSE DENY ***
60    PRINT
65    PRINT «PRESS LETTER A TO D»
70    A$ = INKEY$
80    IF A$ = «A» OR A$ = «B» OR A$ = «C» OR A$ = «D» THEN
      PRINT «O.K» ELSE GOTO 70
90    IF A$ = «A» THEN GOTO 200 ELSE PRINT «NOT A»
100   IF A$ = «B» THEN GOTO 300 ELSE PRINT «NOT B»
110   IF A$ = «C» THEN 400 ELSE PRINT «NOT C»
120   IF A$ = «D» THEN 500 ELSE PRINT «NOT D»
130   REM *** NOTE GOTO MAY BE OMMITTED ***
140   REM *** AS IN LINES 110 AND 120 ***
200   FOR X = 1 TO 20
210   PRINT «A»;
220   NEXT X
230   GOTO 60
300   FOR X = 1 TO 10
310   PRINT «B»;
320   NEXT X
330   GOTO 60
400   FOR X = 1 TO 10
410   PRINT «C»;
420   NEXT X
430   GOTO 60
500   FOR X = 1 TO 10
```

```
510   PRINT «D»;
520   NEXT X
530   GOTO 60
```

لاحظ أننا استغنينا عن البلاغ «GOTO» بعد البلاغ «THEN»، ولم نتمكن من الاستغناء عن غيره من الإيعازات. كذلك لاحظ أنه هناك أربعة اختيارات هي «A, B, C و D». فإذا ما أدخلنا أحد هذه الرموز إلى الحاسب، فإنه سيطبع لنا عبارة «O.K» ويستمر البرنامج. أما إذا لم يكن الرمز الذي أدخلناه إلى الحاسب هو أحد هذه الرموز، فإن الحاسب يهمله ويعود إلى السطر 70 للسؤال ثانية عن رمز جديد.

البرنامج (A34) التالي، مثال آخر على استخدام البلاغ.

```
5     REM **** A34     *****
10    REM *** IF-THEN-ELSE STATEMENT ***
20    REM *** IF STATEMENT IS TRUE ***
30    REM *** THEN ACTION ***
40    REM *** ELSE ***
50    INPUT A
60    IF A > 10 THEN GOTO 70 ELSE GOTO 80
70    PRINT «A IS LARGE»: GOTO 50
80    PRINT «A IS LOW»
90    GOTO 50
```

*** RANDOM

*** عشو

تستخدم الدالة «RND عشو» للحصول على رقمٍ عشوائي، وهذا مفيد جداً في الألعاب المستخدمة مع الحاسب، وصيغة الدالة تكون كالآتي:

```
A = RND(1)
```

هذا يعطي لنا رقماً عشوائياً بين «الصفر» والـ «0.999 9999». ولأجل الحصول على رقم عشوائي بين رقمين محددين استخدم القانون الآتي:

(اطرح الرقم الصغير من الرقم الكبير، أضف واحداً إلى الناتج، اضرب الناتج بـ «RND(1)»، أضف الرقم الصغير إلى الناتج، أهمل الأرقام بعد الفارزة إن وجدت).

مثـــال:

للحصول على رقم عشوائي بين 5 و 20

20 − 5 + 1 = 16

A = INT (RND(1)*16) + 5

استخـدمنا هنـا الدالـة «INT صح» للحصـول عـلى عـدد صحيـح «INTEGER» (أي إهمال الأرقام بعد الفارزة).

مثـــال:

INT (5.4325645435) = 5

البرنامج (A35) أدناه يستخدم الدالـة «RND» للحصول عـلى أرقام عشوائية بين «الصفر» والـ «0.9999999».

```
5     REM **** A35     ****
10    REM *** RANDOM STATEMENT ***
20    REM *** RND(1) RETURNS A ***
30    REM *** RANDOM NUMBER FROM ***
40    REM *** 0 TO 0.999999      ***
50    FOR X = 1 TO 10
60    A = RND(1)
70    PRINT A
80    NEXT X
```

أما البرنامج (A36) التالي فيعطي أرقاماً عشوائية بين «1» و «5».

```
5     REM **** A36     ****
10    REM **** RANDOM STATEMENT ***
20    REM **** RND(1) RETURNS A
30    REM **** RANDOM NUMBER FROM
40    REM **** 0 TO 0.999999
50    REM **** INTEGER TO REMOVE
60    REM **** DECIMALES
70    FOR X = 1 TO 100
'80   A = INT (RND(1) * 5) + 1
90    PRINT A
100   NEXT X
```

يستخدم البرنامج (A37) الدالة «RND» للسيطرة على البلاغ «LOCATE» للحصول على أشكال على الشاشة. الأسطر من 60 إلى 80 تعطي مجموعة من الأرقام العشوائية بين «الصفر و «29» وبين «الصفر» و «22». واستخدم الأمر «PRINT» في السطر 90 ليطبع الرمز «0» عشوائياً، حسب المواقع التي حددت له باستخدام البلاغ «LOCATE». أما الأسطر من 130 إلى 170 فتحدد مواقع عشوائية لتطبع فراغاً فيها. (إذا ما وقع الفراغ على أحد الرموز السابقة «0» فسيمسحها).

```
5    REM **** A37    ****
10   REM *** RANDOM STATEMENT ***
20   REM *** LOCATE FOR-NEXT **
30   REM *** DISPLAY PROGRAM TO ***
40   REM *** SHOW USE OF ***
50   REM *** COMMANDS ***
60   FOR A = 1 TO 1000
70   X = INT (RND(1) * 30)
80   Y = INT (RND(1) * 23)
90   LOCATE X, Y: PRINT «0»
100  NEXT A
110  REM *** RND FOR CLEAR ***
120  REM *** CLEAR CENTRE ONLY ***
130  FOR A = 1 TO 1000
140  X = INT (RND(1) * 10) + 10
150  Y = INT (RND(1) * 10) +5
160  LOCATE X, Y: PRINT «  »
170  NEXT A
```

*
**

الفصل الثامن
فن التشكيل

سندرس في هذا الفصل فن التشكيل في الحاسبات (GRAPHICS)، وهذا هو الاسم الذي يطلق على أي عرض لأشكال معينة أو جزء منها، وله رموز خاصة به من أرقام وحروف وعلامات وصل.

بعض الإيعازات المستخدمة في هذا الفصل موضحة بشكل وافٍ في الفصل السابع، لذلك سوف لن نستعرضها بشكل مفصل في هذا الفصل.

يقسم فن التشكيل في حاسبات (M.S.X.) إلى ثلاثة أقسام:

1 ـ رموز تخطيطية (نقشية): (GRAPHICS CHARACTERS)

ويكون التشكيل فيها باستخدام رموز موجودة في لوحة المفاتيح مثل الحروف والأرقام، ويمكن ترتيبها على الشاشة لإعطاء أشكال ونقوش معينة.

2 ـ النقوش ذات الدقة العالية: (HI-RES. GRAPHICS)

ويكون التشكيل فيها باستخدام الشاشة ذات الدقة العالية.

3 ـ رسوم شبحية: (SPRITE GRAPHICS)

وهذه تكون نقشات صغيرة وذات دقة عالية، وجاهزة بحيث يمكن تحريكها بسهولة على الشاشة دون التأثير أو الاعتماد على أسلوب التشكيل. ويمكن عرضها على كلتا الشاشتين العادية وذات الدقة العالية.

الأمثلة التالية توضح كيفية الحصول على هذه النقشات واستخدامها، وفي البداية من الأفضل فهم أسلوب استخدام الشاشات الأربع المختلفة في حاسبات (M.S.X.). وهي 0, 1, 2 و3، وخصائصها كما يلي:

الشاشة 0:

وهي شاشة اختبارية، تستخدم في البرامج العامة وأبعادها «40 × 24» رمزاً.

الشاشة 1:

وهي شاشة اختبارية أيضاً، وتستخدم في البرامج العامة، وأبعادها «32 × 24» رمزاً.

الشاشة 2:

وهي الشاشة ذات الدقة العالية، وتستخدم لأغراض التشكيل والرسم وأبعادها «255 × 191» رمزاً.

الشاشة 3:

وهي الشاشة ذات الدقة المتوسطة، والألوان المتعددة. وتستخدم لأغراض التشكيل والرسم، وأبعادها «64 × 48» رمزاً.

*** شاشة *** SCREEN

المثال التالي تطبيق بسيط للنوع الأول من أنواع التشكيل، حيث يقوم البرنامج (A38) التالي بأشغال الشاشة بالرمز «H»، حيث السطر 30 يحدد أسلوب العرض (DISPLAY MODE) وذلك باختيار الشاشة «1» «SCREEN 1». السطر 40 يحدد عرض الشاشة بـ 32 رمز فقط.

السطر 50 يقوم بإخفاء الكتابة الموجودة أسفل الشاشة والتي تمثل عمل مفاتيح الدوال، وهذا إيعاز عام يمكن استخدامه ضمن أي برنامج وكذلك يمكن استخدامه مستقلاً.

السطر 60 يحمل ملحوظة (REM)، تبين أن الخطوات التي ستلي هذا السطر هي المسؤولة عن ملء الشاشة بالرمز المطلوب.

الأسطر من 60 إلى 120 تحمل دواري (FOR-NEXT) عملهما الأساسي

هو تحديد المكان الذي سيطبع فيه الرمز ومن ثمة طباعته، وفي كل دورة تحدد إحداثيات مكان طبع الرّمز «H». (حاول دراسة خاصية هذين الدوارين).

أما السطر 130 فإنه يثبت الشكل على الشاشة.

بعد أن تأكدت من طباعة البرنامج بصورة صحيحة أعطِ الأمر «نفذ RUN». لاحظ الآن أن الشاشة قد امتلأت بالرمز «H» من الاحداثي «0,0» إلى الإحداثي «31, 23».

```
5     REM **** A38     ****
10    REM *** SCREEN SIZE ***
20    REM *** SET SCREEN SIZE AND WIDTH ***
30    SCREEN 1
40    WIDTH 32
50    KEY OFF
60    REM *** FILL SCREEN
70    FOR CLM = 0 TO 31
80    FOR ROW = 0 TO 23
90    LOCATE CLM, ROW
100   PRINT «H»;
110   NEXT ROW
120   NEXT CLM
130   GOTO 130
```

لإرجاع الشاشة إلى الوضع الطبيعي، اضغط على المفتاحين (CTR + STOP) في آن واحد، وذلك لإيقاف البرنامج. ثم أعطِ البلاغين التاليين:

```
SCREEN 1
WIDTH 29
```

غير الآن السطر 30 ليستخدم الحاسب الشاشة 0 في تنفيذ البرنامج، وبذلك يصبح البرنامج كالآتي:

```
5     REM **** A38     ****
10    REM *** SCREEN SIZE ***
20    REM *** SET SCREEN SIZE AND WIDTH ***
```

```
30    SCREEN 0
40    WIDTH 32
50    KEY OFF
60    REM *** FILL SCREEN
70    FOR CLM = 0 TO 31
80    FOR ROW = 0 TO 23
90    LOCATE CLM, ROW
100   PRINT «H»;
110   NEWT ROW
120   NEXT CLM
130   GOTO 130
```

بعد تنفيذ البرنامج يمكن لك أن تلاحظ أن العرض سيكون في مركز الشاشة تقريباً، حيث حدد السطر 30 الشاشة بأبعاد «24 × 40» ولكن السطر 40 حدد عرض الشاشة بـ 32 رمزاً فقط، لذلك سيكون العرض من الإحداثي «0,0» إلى الإحداثي «31, 23».

غير الآن السطر 40 ليصبح البرنامج كما يلي:

```
5     REM **** A39     ****
10    REM *** SCREEN SIZE ***
20    REM *** SET SCREEN SIZE AND WIDTH ***
30    SCREEN 0
40    WIDTH 40
50    KEY OFF
60    REM *** FILL SCREEN ***
70    FOR CLM = 0 TO 31
80    FOR ROW = 0 TO 23
90    LOCATE CLM, ROW
100   PRINT «H»;
110   NEXT ROW
120   NEXT CLM
130   GOTO 130
```

السطر 40 يحدد العرض الكامل للشاشة، وبذلك فإن مساحة العرض ستكون أكبر مما يحتاجه البرنامج.

تعلمنا مما سبق أهم البلاغات المستخدمة في فن التشكيل وهما

«WIDTH, SCREEN». اطبع البرنامج (A40) التالي ونفذه ولاحظ تأثير البلاغين السابقين.

```
5     REM **** A40     ****
10    REM **** SCREEN SIZE
20    REM **** SET SCREEN SIZE AND WIDTH
30    SCREEN 1
40    WIDTH 5
50    KEY OFF
60    REM **** FILL SCREEN
70    FOR X = 1 TO 80: PRINT «THE M.S.X. COMPUTER»
80    BEEP
90    NEXT X
100   GOTO 100
```

*** COLOR *** لـــون

سندرس الآن تأثير الألوان على البرامج (انتبه جيداً فإننا نستخدم كلمة COLOR الامريكية وليس COLOUR الانكليزية).

إذا ما نظرنا إلى الشاشة نلاحظ هناك ثلاث مناطق مختلفة الألوان وهي:

1 ـ الخلفية: (BACKGROUND)

ويحدد لونها حسب البلاغ «SCREEN» وتكون بلون أزرق في الحالة الاعتيادية ويقصد بها لون الخلفية للشاشة.

2 ـ الكتابة: (FOREGROUND)

ويقصد به لون الأحرف والرموز، ويحدد كسابقة.

3 ـ الإطار: (BORDER)

ويمثل لون المساحة المحيطة بمنطقة العرض، وتكون بلون أزرق فاتح في الحالة الاعتيادية.

يتبع البلاغ «COLOR» بثلاث عناصر (أرقام) سبق التطرق لها في الفصل، وكل عنصر عبارة عن رمز اللون المخصص لأحد المناطق الثلاث المبينة أعلاه. أدناه الألوان والأرقام الممثلة لها:

0	شفاف	TRANSPERENT
1	أسود	BLACK
2	أخضر متوسط	MEDIUM GREEN
3	أخضر فاتح	LIGHT GREEN
4	أزرق غامق	DARK BLUE
5	أزرق فاتح	LIGHT BLUE
6	أحمر غامق	DARK RED
7	أزرق غامق جداً	CYAN
8	أحمر متوسط	MEDIUM RED
9	أحمر فاتح	LIGHT RED
10	أصفر غامق	DARK YELLOW
11	أصفر فاتح	LIGHT YELLOW
12	أخضر غامق	DARK GREEN
13	أحمر مزرق	MAGENTA
14	رصاصي	GREY
15	أبيض	WHITE

اطبع الآن الإيعاز التالي:

```
COLOR 1, 9, 3 (CR)
```

ستلاحظ تلون الشاشة بالألوان المحددة في الإيعاز، راجع الألوان مع الجدول السابق.

البرنامج (A41) التالي يتضمن برنامج تأخير (DELAY PROG.) ويستخدم هذا للإقلال من سرعة تنفيذ البرنامج. وبرنامج التأخير هذا ما هو إلا دوار (FOR-NEXT)، عدد دوراته يحدد زمن التأخير. مثلاً:

```
FOR DLY = 1 TO 100
NEXT DLY
```

في هذا الدوار تم التأخير بـ 100 دورة فإذا ما كانت كل دورة تؤخر 1/100 من الثانية (مثلاً). فإن هذا الدوار سوف يؤخر البرنامج لمدة ثانية واحدة، وهكذا. تكون الحاجة ملحة لهذا البرنامج في مجال العروض لكي نجعل الصورة أو الشكل ثابت بعض الوقت على الشاشة.

في البرنامج (A41) التالي ثلاث دوارات بثلاث متغيرات لكل من الألوان الثلاثة: الخلفية، الإطار، والرموز. الأسطر من 70 إلى 120 تجعل الحاسب يطبع لك رسالةتذكرك بخصائص التلوين وتنبهك إلى وجود 4096 حالة ستظهر أمامك. السطر 130 يحوي ملاحظة تشير إلى أن الجزء الذي يليه من البرنامج يقوم باختيار الألوان. أما الدوارات الخاصة بتغيير الألوان فهي محصورة بين السطرين 130 و 230 بضمنها برنامج التأخير الذي يشمل الخطوات من 175 إلى 200. بعد انتهاء البرنامج تعود الشاشة إلى التلوين الطبيعي لها، وفقاً للبلاغ الموجود في السطر 240. البرنامج طويل بعض الشيء لكثرة عدد الاحتمالات، لذلك يمكنك الضغط على (CTRL + STOP) في آن واحد لإيقاف البرنامج. ولإعادة الألوان الطبيعية للشاشة اضغط على مفتاح الدوال «F6» أي المفتاح (SHIFT + F1).

```
5     REM *** A41 ***
10    REM *** X, Y, Z = FOREGROUND ***
20    REM *** BACKGROUND, BORDER COLORS ***
30    REM *** DISPLAY TEXT COLORS ***
40    REM *** SET TO THE TEXT MODE ***
50    SCREEN 1
60    REM *** PRINT DISPLAY
70    LOCATE 0,10
80    PRINT «FOREGROUND, BACKGROUND, BORDER»
90    LOCATE 0, 14
100   PRINT «SEE THE COLOURS CHANGE»
110   LOCATE 0, 18
120   PRINT «4096 COMBINATIONS»
130   REM *** LOOP FOR COLOUR OPTIONS»
140   FOR X = 0 TO 15
150   FOR Y = 1 TO 15
160   FOR Z = 2 TO 15
```

```
170  COLOR X, Y, Z
175  REM *** DELAY DISPLAY
180  REM *** DELAY DISPLAY
190  FOR DLY =1 TO 100
200  NEXT DLY.
210  NEXT Z
220  NEXT Y
230  NEXT X
240  REM *** RESET NORMAL COLOUR
250  COLOR 15, 4, 7
```

*** PSET, PRESET *** نقطة، لانقطة

نحاول الآن استخدام البلاغين «PSET, PRESET» مع الشاشة العالية الدقة لوضع الأشكال المعينة والأشباح في مناطق محددة من الشاشة. مهمة البلاغ «PSET» هي وضع نقطة محددة على الشاشة، إحداثياتها يمكن أن تكون من الإحداثي (0,0) في أقصى الجهة العليا للشاشة إلى الإحداثي (191, 255) في أقصى الجهة السفلى للشاشة. الصيغة العامة للبلاغ هي:

PSET (N, M), C

حيث تمثلان (N, M) إحداثيات النقطة، وتمثل (C) لونها (راجع جدول الألوان السابق). أما إذا ما استخدم البلاغ دون تحديد اللون، فستظهر النقطة باللون الذي يحدده الحاسب وغالباً يكون اللون الخاص بالرموز (FOREGROUND). فإذا ما أردنا وضع نقطة باللون الأحمر الغامق مثلاً وفي منتصف الشاشة يكون البلاغ بالصيغة التالية:

PSET (128, 95), 6

حدد الآن نقاط معينة وبألوان تختارها لترسم أشكال معينة على الشاشة. البلاغ «PRESET» يستخدم بالطريقة السابقة نفسها ما عدا أنه يرجع لون النقطة المحددة إلى اللون الطبيعي للشاشة (لون الخلفية BACKGROUND). هذا فيما إذا لم يحدد لون معين في صيغة البلاغ. أما إذا حدد لون معين فإن عمله سيكون مشابهاً لعمل البلاغ «PSET».

البرنامج (A42) التالي يقوم برسم نقاط على الشاشة بشكل عشوائي، حيث يحدد السطر 30 نوع الشاشة، بالشاشة ذات الدقة العالية (SCREEN 2). والسطرين 40, 50 يحددان رقمين عشوائيين باستخدام الدالة (عشو RND)، وهي تحدد ذلك ضمن الإحداثيين (0, 0), (191, 255). هذه فرصة جيدة لدراسة الدالة «RND».

لاحظ المعادلة التالية:

```
X = INT (RND(1) * 255)
```

هذه سوف تجعل «X» يتغير عشوائياً بين الرقمين (0) و (255). أما المعادلة:

```
Y = INT (RND(1) * 191)
```

فإنها تجعل «Y» تتغير عشوائياً بين الرقمين (0) و (191). الدالة (صح INT) فهي لأخذ الأعداد الصحيحة فقط لقيم «X» و «Y». أما السطر 60 فيحمل البلاغ «PSET» الذي يضع نقطة على الشاشة، حسب الإحداثيات التي تحددها المعادلتين السابقتين.

السطر 70 يجعل الحاسب يدور في دوار مستمر، إلى أن يوجه له الإيعاز بالتوقف (STOP + CTRL). بعد تنفيذ البرنامج، أوقفه وأبدل السطر 30 ليصبح كالآتي:

```
30  SCREEN 3
```

ولاحظ الفرق بين خصائص الشاشتين.

```
5   REM **** A42     *****
10  REM *** (X,Y) = SCREEN CO-ORDINATES ***
20  REM *** SET RANDOM POINTS ON SCREEN ***
30  SCREEN 2
40  X = INT(RND(1) * 255)
50  Y = INT (RND(1) * 191)
60 PSET (X,Y)
70  GOTO 40
```

البرنامج التالي (A43) يستخدم البلاغ «PSET» لرسم خط عمودي على الشاشة وبذلك يجب أن نثبت مقدار «X» ونغير مقدار «Y».

الخطوات من (90) إلى (110) تمثل دوار يغير مقدار «Y» من (95) إلى (191) ثم يرسم ذلك على الشاشة في السطر 100. السطر 70 يثبت مقدار «X» عند 128. والسطر 130 يثبت الشكل على الشاشة.

البرنامج يؤدي ذلك بسرعة، لذلك من الأفضل إضافة برنامج التأخير الذي تم توضيحه سابقاً لإتاحة الإمكانية لمتابعة البرنامج.

```
5   REM *** A43 ***
10  REM *** (X,Y) = SCREEN CO-ORDINATES ***
20  REM *** DRAW VERTICAL LINE ***
30  REM *** USING PSET ***
40  REM *** SET HI-RES MODE ***
50  SCREEN 2
60  REM *** FIX X CO-ORDINATES ***
70  X = 128
80  REM *** PRODUSE Y CO-ORDINATE ***
90  FOR Y = 95 TO 191
100 PSET (X,Y)
110 NEXT Y
120 REM *** HOLD SCREEN DISPLAY ***
130 GOTO 130
```

البرنامج الآخر (A44) يثبت قيم «Y» في السطر 70 ويغير قيم «X» في الأسطر (110, 100, 90).

```
5   REM **** A44     *****
10  REM *** (X, Y) = SCREEN CO-ORDINATES ***
20  REM *** DRAW HORIZONTAL LINE ***
30  REM *** USING PSET ***
40  REM *** SET HI-RES MODE ***
50  SCREEN 2
60  REM *** FIX Y CO-ORDINATES ***
70  Y = 95
80  REM *** PRODUCING X CO-ORDINATES ***
90  FOR X = 128 TO 255
100 PSET (X,Y)
```

```
110   NEXT X
120   REM *** HOLD SCREEN DISPLAY ***
130   GOTO 130
```

لاحظنا في المثالين السابقين كيفية تغيير مقدار كل من «X» و«Y»، وبالتأكيد لو غيرنا مقداريهما بالنسبة نفسها وفي برنامج واحد، فإن الرسم الناتج يكون عبارة عن خط مستقيم يقطع الشاشة قطرياً. (حاول ذلك).

البرنامج التالي (A45) يستخدم البلاغ السابق لرسم خطوط عمودية، حيث يغير «Y» مرحلياً، بعد أن يغير «X» لكل قيمة من قيم «Y» لرسم خط مستقيم عمودي، لذلك نلاحظ بأن الأسطر 110, 100, 90 تحوي دوار يرسم خط عمودي لقيمة «Y» المحددة. بعد ذلك يقلل السطر 130 من قيمة «Y». وأما السطر 150 فإنه يعود بالحاسب إلى السطر 80 ليعيد الرسم مرة أخرى ولقيمة جديدة لـ «Y».

```
5     REM **** A45     *****
10    REM *** (X,Y) = SCREEN CO-ORDINATE
20    REM *** DRAW HORIZONTAL LINE ***
30    REM *** USING PSET ***
40    REM *** SET HI-RES MODE
50    SCREEN 2
60    REM *** FIX Y CO-ORDINATES
70    Y = 95
80    REM *** PRODUCE X CO-ORDINATES
90    FOR X = 128 TO 225
100   PSET (X,Y)
110   NEXT X
120   REM *** DECREMENT Y CO-ORDINATE
130   Y = Y - 3
140   REM *** CHECK NOT OFF SCREEN ***
150   IF Y < 0 THEN GOTO 190
160   REM *** REPEAT LINE AT NEW POSITION
170   GOTO 80
180   REM *** HOLD SCREEN DISPLAY **
190   GOTO 190
```

*** سطـــر *** LINE

يستخدم هذا البلاغ لرسم خط مستقيم في أي مكان على الشاشة، وبما أنه الخط المستقيم يرسم بين نقطتين، لذلك فإن الصيغة العامة للبلاغ تكون كالآتي:

LINE (N1, M1) – (N2, M2)

حيث تمثل (N1, M1) إحداثيات النقطة الابتدائية للمستقيم، و (N2, M2) إحداثيات النقطة النهائية له.

وهذا البلاغ يستخدم خاصية البلاغ «PSET» نفسها في الرسم، حيث يرسم الخط المستقيم على شكل نقاط متجاورة. مثال:

LINE (0,0) – (225, 191)

هذا الإيعاز يرسم خط مستقيم من الزاوية العليا اليسرى للشاشة إلى الزاوية السفلى اليمنى. ولون الخط سيكون بلون الرموز (FOREGROUND)، حيث لم نخصص لون معين للرسم في الإيعاز. لا تنسى العلامة « – » في الإيعاز (علامة السالب في الحاسب)، وهذه تعني «إلى». حاول التدرب على ذلك بإعطاء إيعازات أخرى مشابهة للإيعاز السابق. اطبع ونفذ البرنامج التالي (A46) والذي يرسم لك خط مستقيم يقطع الشاشة.

```
10   REM **** A46      *****
20   REM *** DRAW LINES USING THE ***
30   REM *** LINE COMMAND ***
40   REM *** SET HI-RES MODE ***
50   SCREEN 2
60   REM *** SET START AND END ***
70   LINE (255, 0) – (0, 191)
80   REM *** HOLD SCREEN DISPLAY ***
90   GOTO 90
```

الآن حاول استخدام البلاغ «LINE» لرسم المحاور التي تم رسمها باستخدام البلاغ «PSET» في البرامج السابقة، وستلاحظ أن البرامج ستكون أسهل بكثير مما سبق.

استخدم البرنامج (A47) البلاغ «PSET» لوضع نقاط في إحداثيات محددة ثم البلاغ «LINE» لرسم خطوط مستقيمة بينها.

```
10    REM ** A47 **
80    SCREEN 2
90    COLOR 1, 7, 7
100   CLS
110   DIM A(12), B(12)
120   FOR N = 1 TO 12
130   K = N/6 * 3.1415927 #
140   A(N) = 128 + 80 * SIN(K): B(N) = 8 + 80 * COS(K)
150   PSET (A(N), B(N) )
160   NEXT N
170   FOR N = 1 TO 12
180   FOR M = 1 TO 12
190   LINE (A(N), B(N) ) - (A(M), B(M) )
200   NEXT M
210   NEXT N
220   GOTO 220
```

من الخصائص الأخرى للبلاغ «LINE» هي إمكانية تحديد لون الرسم وذلك بوضع رمز يدل على اللون المطلوب، في نهاية البلاغ، لتصبح الصيغة كالآتي:

```
LINE (N1, M1) - (N2, M2), C
```

حيث «C» تحدد لون المستقيم. لنستخدم الآن «C» ونجعل مقدارها يتغير عشوائياً، كما في البرنامج (A48) التالي.

```
5     REM **** A48     *****
10    REM *** X, Y = START CO-ORDINATE
20    REM *** A, B = END CO-ORDINATE
30    REM *** LP = LOOP
40    REM *** DRAW RANDOM LINES
50    REM *** USING LINE COMMAND
60    REM *** SET HI-RES MODE
70    SCREEN 2
80    FOR LP = 1 TO 500
90    REM *** SET START CO-ORDINATE
```

```
100  X = INT(RND(1) * 256)
110  Y = INT(RND(1) * 191)
120  REM *** SET END CO-ORDINATE
130  A = INT(RND(1) * 256)
140  B = INT(RND(1) * 191)
150  REM *** SELECT COLOUR
160  C = INT(RND(1) * 16)
170  REM *** DRAW LINE
180  LINE (X, Y) - (A, B), C
190  NEXT LP
200  REM *** HOLD DISPLAY ON SCREEN
210  GOTO 210
```

لاحظ أن الألوان قد امتجزت وتداخلت، وهذا بسبب استخدام الشاشة ذات الدقة العالية. لذلك غير السطر 80 ليصبح بالشكل التالي:

```
80  SCREEN 3
```

وهذا سوف يؤدي إلى استخدام الشاشة ذات الألوان المتعددة. البرنامج يقوم برسم 500 خط ويستغرق ذلك فترة طويلة، لذلك يمكنك الضغط على المفتاحين (CTRL + STOP) لإيقاف البرنامج. لإيقاف البرنامج لفترة واستمراره بعد ذلك اضغط على المفتاح (STOP) فقط، وعند الضغط عليه مرة أخرى يستمر البرنامج.

يستخدم البلاغ «LINE» كذلك لرسم شكل مربع مباشرة دون الحاجة إلى رسم كل ضلع فيه منفرداً، وذلك بتحديد نقطتين، الخط الواصل بينهما يمثل قطر المربع. وبإضافة الرمز «B» بعد الرمز الدال على اللون، لتكون الصيغة كالآتي:

```
LINE (N1, M1) - (N2, M2), C, B
```

اطبع المثال التالي:

```
LINE (0,0) - (255, 191), 9, B
```

ستلاحظ مستطيلاً أحمراً يحيط بالشاشة. أما إذا أردنا رسم المستطيل بلون يختاره الحاسب، وعادة ما يكون لون الرموز (FORGROUND)، نترك مكان

اللون فارغاً، مع الانتباه بترك الفارزة المخصصة له في مكانها، لتصبح الصيغة كالآتي:

```
LINE (0,0) – (255, 191),, B
```

```
5     REM **** A49      *****
40    REM *** DRAW A RECTANGLE ***
50    REM *** SET HI-RES MODE ***
60    SCREEN 2
70    REM *** SET UP PARAMETERS ***
80    X = 50: Y = 50
90    A = 200: B = 150
100   C = 9
110   REM *** DRAW LINE
120   LINE (X, Y) – (A,B), C
130   REM *** DRAW RECTANGLE
140   LINE (X, Y) – (A, B), C, B
150   REM *** HOLD DISPLAY ON SCREEN
160   GOTO 160
```

يستخدم البرنامج السابق (A49) البلاغ «LINE» في مهمتين: الأولى لرسم خط مستقيم، كما في الأمثلة السابقة؛ والثانية لرسم مربع متكامل كما ورد في السطر 140. لاحظ اختلاف الإحداثيات في الحالتين. حاول الآن طباعة برنامج يرسم عدد من المربعات في شاشة واحدة.

يستخدم البلاغ «LINE» كذلك لرسم مساحات ملونة على شكل مربع وذلك بمجرد وضع الرمز «BF» بدلاً من «B»، وبذلك تكون الصيغة العامة كالآتي:

```
LINE (N1, M1) – (N2, M2), C, BF
```

اطبع الخطوة التالية:

```
LINE (0,0) – (255, 191), 9, BF
```

لاحظ أن الشاشة قد امتلأت بمستطيل أحمر اللون يغطي الإحداثيات (0,0) إلى (255, 191).

وكما تعلمنا يمكن ترك مكان الرمز الخاص باللون فارغاً، ليملء المستطيل باللون المخصص للرموز (FORGROUND)، على أن لا ننسى الفارزة المخصصة للون. اطبع الإيعاز التالي ولاحظ الفرق:

```
LINE (0,0) - (255, 191),, BF
```

البرنامج (A50) أدناه يوضح استعمال البلاغ «LINE» لرسم مستطيلات ملونة. لاحظ أن الرسم يكون كإطار خارجي فقط ثم يملء باللون المحدد.

الأسطر من 80 إلى 100 تعرّف المتغيرات، لمواقع الإحداثيات. السطر 130 يرسم المستطيل ويملؤه باللون المحدد. أما السطر 150 فيثبت الشكل على الشاشة.

```
5   REM **** A50     *****
30  REM *** DRAW A BOX AND ***
40  REM *** FILL WITH COLOUR ***
50  REM *** HI-RES SCREEN
60  SCREEN 2
70  REM *** SET UP PARAMETERS
80  X = 50: Y = 50
90  A = 200: B = 150
100 C = 3
110 REM *** DRAW RECTANGLE
120 REM *** AND FILL
130 LINE (X,Y) - (A,B), C, BF
140 REM *** HOLD DISPLAY ON SCREEN
150 GOTO 150
```

حاول الآن رسم أكثر من مستطيل وفي مواقع مختلفة، حدد إحداثيات مختلفة واجعل الرسومات متداخلة، ثم استخدم الشاشة 3 (SCREEN 3) ولاحظ الفرق كذلك. استخدم ألوان مختلفة لتتمرن جيداً على استخدام البلاغ.

*** CIRCLE

*** دائرة

البلاغ «CIRCLE» يمكننا من رسم الأشكال الدائرية، وكما في الإيعازات الأخرى الخاصة بالرسم يستخدم البلاغ مع الشاشتين (3, 2) فقط. وتكون صيغته كالآتي:

```
CIRCLE (N, M), D
```

حيث تمثل (N, M) إحداثيات مركز الدائرة، وتمثل «D» نصف قطرها. (تكون جميع القياسات بعدد لنقاط). اطبع الآن المثال التالي:

```
CIRCLE (128, 95), 20
```

لاحظ أن الإيعاز يجعل الحاسب يرسم دائرة مركزها في منتصف الشاشة تقريباً، ونصف قطرها يساوي 20 نقطة، وبلون الرموز. استخدم البلاغ «CLS» قبل طباعة الإيعاز أعلاه لتنظيف الشاشة مما كتب عليها سابقاً.

اطبع البرنامج (A51) أدناه ولاحظ أنه سيقوم برسم دائرة في الشاشة 2 (ذات الدقة العالية). الأسطر من 70 إلى 80 تحدد مركز ونصف قطر الدائرة. السطر 100 يرسم الدائرة، حسب الإحداثيات السابقة. وأما السطر 120، فيحفظ الشكل على الشاشة.

```
5     REM **** A51     *****
10    REM *** X, Y = POSITION
20    REM *** R = RADIUS
30    REM *** DRAW A CIRCLE ***
40    REM *** SET HI-RES MODE
50    SCREEN 2
60    REM *** SET UP PARAMETERS
70    X = 125: Y = 95
80    R = 50
90    REM *** DRAW CIRCLE
100   CIRCLE (X, Y), R
110   REM *** HOLD DISPLAY ON SCREEN
120   GOTO 120
```

ستلاحظ في المثال السابق بأن الشكل ليس دائرياً تماماً، وذلك بسبب

تصميم شاشة جهاز العرض (التلفزيون)، ويختلف التشوه باختلاف الجهاز، ويمكن التغلب عليه باستخدام رمز آخر (سنأتي عليه لاحقاً).

يمكن لنا أن نحدد لون الرسم مع البلاغ «CIRCLE» كما حددناه مع البلاغ «LINE» وذلك بوضع الرمز الخاص باللون بعد الرمز الخاص بالقطر، وتكون الصيغة كالآتي:

```
CIRCLE (M, N), D, C
```

اطبع المثال التالي، ولاحظ أن الدائرة سترسم بحيث يكون مركزها في الإحداثيات (50, 50) ونصف قطرها 10 نقاط، وباللون البنفسجي.

```
CIRCLE (50,50) .10,13
```

البرنامج (A52) التالي يوضح عملية رسم دائرة بلون محدد.

```
5    REM **** A52      *****
10   REM *** X, Y = POSITION
20   REM *** R = RADIUS
30   REM *** DRAW A CIRCLE ***
40   REM *** SET HI-RES MODE
50   SCREEN 2
60   REM *** SET UP PARAMETERS
80   X = 125: Y = 95
90   R = 50: C = 1
100  REM *** DRAW CIRCLE
110  CIRCLE (X,Y), R
120  CIRCLE (X,Y), R, C
130  REM *** HOLD DISPLAY ON SCREEN
140  GOTO 140
```

كما نعلم أن الشكل الدائري ما هو إلا حالة خاصة من الشكل البيضوي. وفي نظام (M.S.X.) يمكن رسم الأشكال البيضوية باستخدام البلاغ «CIRCLE» وذلك بمجرد وضع رمز في نهاية الصيغة يمثل النسبة بين قطري الشكل البيضوي. وبذلك تصبح الصيغة كالآتي:

```
CIRCLE (M, N), D, C, , , A)
```

حيث تمثل «A» النسبة بين قطري الشكل البيضوي. لاحظ هناك فراغين قد تركا في الإيعاز، سنعود إليهما لاحقاً.

لنرسم الدائرة التالية:

```
CIRCLE (128, 45), 20, 8, , , 1
```

وهذه ستكون باللون الأحمر الفاتح وبنصف قطر مقداره 20 نقطة. والنسبة بين القطرين مقدارها واحد صحيح. وتأثير هذه النسبة يكون عندما لا تساوي واحد، فتكون أكبر من الواحد أو أصغر منه. فإذا ما كانت أصغر من واحد فإن القطر باتجاه «Y» يصبح أصغر، أما إذا كانت النسبة أكبر من واحد فإن القطر باتجاه «X» يكون أصغر.

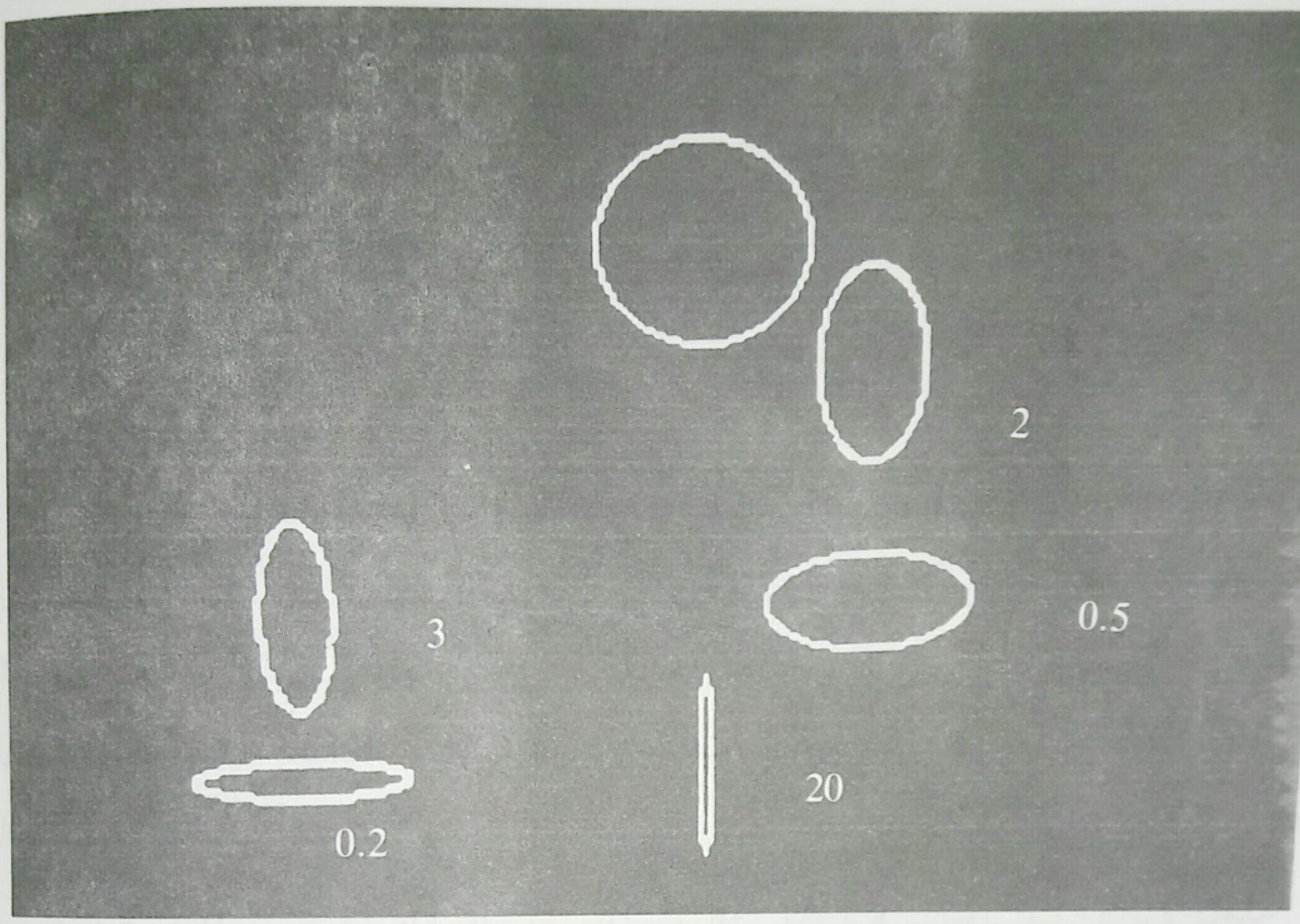
إذا ما افترضنا أن الشكل ذي النسبة التي مقدارها واحد يمثل دائرة كاملة ومنتظمة، فإن النسب التالية تدل على ما يقابلها من أشكال:

2	شكل بيضوي ارتفاعه ضعف عرضه
3	شكل بيضوي ارتفاعه ثلاثة أضعاف عرضه
20	شكل بيضوي ارتفاعه يساوي عشرون ضعفاً من عرضه
0.5	شكل بيضوي عرضه يساوي ضعف ارتفاعه
0.2	شكل بيضوي عرضه يساوي خمسة أضعاف ارتفاعه

المثال التالي يمثل شكل النسبة بين قطريه هي 1.15 وهذا يعطي شكل دائرة أكثر مما لو كانت النسبة 1 ولسبب الذي بيناه سابقاً.

```
CIRCLE (120, 60), 15, , , , 1.15
```

في البرنامج (A53) التالي، السطر 120 يرسم الدائرة بلون الرموز وبنسبة مقدارها واحد بين القطرين، والسطر 130 يرسم الدائرة نفسها، ولكن بنسبة مقدارها (1.5)، وبلون مختلف، ولكن بنفس المنطقة.




```
5     REM **** A53      ****
10    REM *** X, Y = POSITION
20    REM *** R = RADIUS·   C = COLOUR
30    REM *** AS = ASPECT RATIO
40    REM *** DRAW A CIRCLE ***
50    REM *** SET HI-RES SCREEN
60    SCREEN 2
70    REM *** SET UP PARAMETERS
80    X = 125: Y = 95
90    R = 50: C = 1: AR = 1.5)
100   REM *** DRAW CIRCLES
110   REM *** NOTE ASPECT RATIO
120   CIRCLE (X,Y), R
130   CIRCLE (X, Y), R, C, , , AR
140   REM *** HOLD DISPLAY ON SCREEN
150   GOTO 150
```

لاحظ البرنامج (A54) أدناه حيث يرسم السطر 120 دائرة بلون الرموز وبنسبة مقدارها واحد، والسطر 130 يرسم الدائرة نفسها، ولكن بنسبة مقدارها 0.3 وبلون مخالف.

```
5     REM **** A54      *****
10    REM *** X, Y = POSITION
20    REM *** R = RADIUS      C = COLOUR
30    REM *** AS = ASPECT RATIO
40    REM *** DRAW A CIRCLE
50    REM *** SET HI-RES MODE
60    SCREEN 2
70    REM *** SET UP PARAMETERS
80    X = 125: Y = 95
90    R = 50: C = 1: AR = .3
100   REM *** DRAW CIRCLES
110   REM *** NOTE ASPECT RATIO
120   CIRCLE (X, Y), R
130   CIRCLE (X, Y), R, C, , , AR
140   REM *** HOLD DISPLAY ON SCREEN
150   GOTO 150
```

لنـرجع الآن إلى الـرموز التي تـركناهـا، وهذه تمثـل زاوية البـدايـة (START ANGLE) وزاوية النهاية (STOP ANGLE)، والقياس يكون بالزوايا النصف قطرية. وتكون زاوية البداية للدائرة «0» وزاوية النهاية «359» درجة وتكون في الجهة اليمنى. واتجاه الرسم يكون عكس اتجاه حركة عقرب الساعة. وبذلك تمثل جهة الشرق الزاوية «0»، الشمال الزاوية «90»، الغرب الزاوية «180»، والجنوب الزاوية «270» درجة. وهذه يجب أن تحول إلى الزوايا النصف قطرية. ولغرض رسم قوس أو جزء من دائرة، يجب أن نحدد زاويتي البداية والنهاية. أما فيما إذا كانت زاوية البدء أصغر من زاوية الانتهاء، فإن الرسم يكون من الزاوية الصغيرة باتجاه الزاوية الأكبر. فإذا ما كانت زاوية البدء تساوي «0» وزاوية الانتهاء تساوي «180» درجة فإن الرسم يكون في الربع الأيسر من الدائرة. أما إذا كانت زاوية البدء أكبر من زاوية الانتهاء، فإن الرسم يكون من الزاوية الكبيرة إلى الزاويـة «359» درجة، وثم من الـزاوية «0» إلى الـزاوية الصغيرة. مثلاً إذا كانت زاوية البدء «270»، وزاوية الانتهاء «180» درجة فإن الرسم يمثل دائرة ناقصة الربع الثالث منها.

لتحويل الدرجات إلى زوايا نصف قطرية، نضرب مقدار الزاوية في

النسبة الثابتة (ط)، والتي مقدارها (3.142) ونقسم الناتج على (180). مثلاً الزاوية «45» درجة تساوي:

(45 × 3.142) / 180 = 0.7855

أي أن الزاوية «45» درجة تساوي «0.7855» زاوية نصف قطرية.

البرنامج (A55) التالي يرسم لنا دائرة، وثم يرسم قوس من الزاوية «0» باتجاه الزاوية «180» وبلون مختلف. لاحظ أن البرنامج يستخدم الرموز نفسها للدائرة والقوس، مع تحديد بداية ونهاية القوس.

```
5   REM **** A55      *****
10  REM *** X, Y = POSITION
20  REM *** R = RADIUS      C = COLOUR
30  REM *** AS = ASPECT RATIO
40  REM *** EA = END ANGLE
50  REM *** SA = START ANGLE
60  REM *** DRAW A CIRCLE
70  REM *** SET HI-RES MODE
80  SCREEN 2
90  REM *** SET UP PARAMETERS
100 X = 125: Y = 95
110 R = 50: C = 1: AS = 1.15
120 SA = 0: EA = 3.142
130 REM *** DRAW CIRCLE
140 REM *** NOTE START/END ANGLES
150 CIRCLE (X, Y), R
160 CIRCLE (X, Y), R, C, SA, EA, AS
170 REM *** HOLD DISPLAY ON SCREEN
180 GOTO 180
```

السطر 100 يحدد مركز الدائرة، بينما السطر 110 يحدد اللون ونصف القطر. أما السطر 120 فيحدد زاويتي البدء والانتهاء، وهذه طبعاً تكون بالزوايا النصف قطرية. أما السطر 150 فيرسم دائرة باللون المحدد للرموز. والسطر 160 يرسم قوس في الجزء العلوي باللون الأسود. حاول إعادة تنفيذ البرنامج (A55) السابق، وباستخدام الشاشة المتعددة الألوان. وذلك بتغيير السطر 80 ليصبح كالآتي:

```
80  SCREEN 3
```

حاول التدريب على ذلك باستخدام زوايا مختلفة للبدء والانتهاء.

*** اصبغ *** PAINT

يستخدم البلاغ «PAINT» لتلوين الأشكال بألوان محددة، ويتم ذلك بوضع نقطة بلون محدد في وسط الشكل ثم يملأ الشكل ككل بذلك اللون. واستخدام هذا البلاغ يختلف جزئياً من شاشة إلى أخرى (شاشة 2 و 3 فقط)، حيث يجب أن يكون لون إطار الرسم لون الصبغ نفسه، في «شاشة 2»، فإذا ما أردنا أن نصبغ دائرة باللون الأزرق يجب أن ترسم باللون الأزرق أيضاً. أما في الشاشة 3 فليس هناك تحديد للون، فيمكن صبغ دائرة خضراء باللون الأحمر مثلاً. والصيغة العامة للبلاغ «PAINT» تكون بالشكل التالي:

PAINT (N,M), C1, C2

حيث تمثل (N,M) إحداثيات نقطة بداية الصبغ (يجب أن تكون داخل الشكل) و«C1» تمثل لون الصبغ، و «C2» تمثل لون رسم الشكل، وتذكر بأن الصبغ يكون من نقطة البداية إلى الإطار الخارجي، وهذا يعني أن الشكل المراد صبغه يجب أن يكون مغلق، فوجود أي فتحة صغيرة في الشكل تؤدي إل تسرب الصبغة إلى خارج الشكل. كذلك يجب أن نتذكر بأن لون الصبغ يكون نفس اللون المستخدم للرسم عند استخدامنا للشاشة ذات الدقة العالية، فإذا ما استخدمنا لون مخالف، فإن الصبغة ستنتشر على الشاشة ماسحة بذلك الشكل.

البرنامج (A56) التالي يستخدم الشاشة ذات الدقة العالية، حيث يحدد السطران (90, 80) نقطة بداية الصبغ (وهنا مثلت النقطة مركز الدائرة أيضاً)، ونصف القطر ولون الصبغة.

```
5    REM **** A56 *****
10   REM *** X,Y = POSITION
20   REM *** R = RADIUS C = COLOUR
30   REM *** BC = BORDER COLOUR
40   REM *** DRAW AND PAINT CIRCLE
50   REM *** SET HI-RES
```

```
60   SCREEN 2
70   REM *** SET UP PARAMETERS
80   X = 128: Y = 90
90   R = 50: C = 3: BC = 3
100  REM *** DRAW CIRCLE
110  CIRCLE (X, Y), R, C
120  REM *** FILL WITH COLOUR
130  PAINT (X, Y), C, BC
140  REM *** HOLD DISPLAY ON SCREEN
150  GOTO 150
```

لاحظ البرنامج (A57) التالي وتابعه بصورة دقيقة، مستفيداً من الملحوظات الواردة فيه.

```
5    REM **** A57      *****
10   REM *** X, Y = POSITION
20   REM *** R = RADIUS      C = COLOUR
30   REM *** CX, CY SCREEN CENTRE
40   REM *** COLOR EXAMPLE
50   REM ***    SET HI-RES SCREEN
60   SCREEN 2
70   REM *** SET UP PARAMETERS
80   X = 100: Y = 95: A = 140: B = 95
90   CX = 128: CY = 95: R = 50
110  REM *** PAINT SCREEN RED
120  PAINT (CX, CY), 6
130  REM *** DRAW BOX
140  REM *** AND FILL IN YELLOW
150  LINE (20, 50) - (230, 140), 11, BF
160  REM *** DRAW CIRCLES
170  CIRCLE (X,Y), R, C
180  CIRCLE (A, B), R, C
190  REM *** PAINT CENTER AREA
200  PAINT (CX, CY), C
210  REM *** DRAW BOX
220  REM *** AND  FILL IN BLUE
230  LINE (80,80) - (170, 100), 5, BF
240  REM *** HOLD DISPLAY ON SCREEN
250  GOTO 250
```

*** DRAW

*** ارسم

تعلمنا فيما سبق كيفية رسم دائرة، مـربع، أوخط مستقيم، ولكن إذا ما أردنا رسم شكل غير منتظم، لا يمكن لنا استخدام البلاغات السابقة. البلاغ «DRAW» يمكّننا من رسم أي شكل مهما بلغ تعقيده وعدم انتظامه. يمكننا البلاغ «DRAW» من رسم خط مستقيم بأي طول كان وبثمانية اتجاهات ويمكن استخدام أي لون للرسم، ويمكن كذلك تصغير أو تكبير الرسم.

للبلاغ «DRAW» رموز خاصة يتوجب استخدامها معه، وهي:

U الرسم بالاتجاه الأعلى

D الرسم بالاتجاه الأسفل

L الرسم بالاتجاه الياسر

R الرسم بالاتجاه اليمين

E الرسم بالاتجاه الأعلى الأيمن

F الرسم بالاتجاه الأسفل الأيمن

G الرسم بالاتجاه الأسفل الأيسر

H الرسم بالاتجاه الأعلى الأيسر

هذه الرموز يجب أن تتبع بأرقام تمثل طول الخط (المقياس بعدد النقاط) يمكن لنا كذلك الرسم من أي نقطة على الشاشة إلى أي نقطة أخرى، وذلك باستخدام الرمز «M» وهذا يجب أن يتبع برقمين يدلان على الإحداثي. مثلاً:

M0, 0 يرسم إلى الجهـة العليا اليسرى من الشاشة.

M0, 191 يرسم إلى الجهة السفلى اليسرى من الشاشة.

M255, 0 يرسم إلى الجهة العليا اليمنى من الشاشة.

وكما بينا يمكن لنا تصغير أو تكبير الرسم، ولأجل ذلك يستخدم الرمز «S» وهذا يتبع برقم يمثل المقياس، والمقياس الحقيقي يمثل برقم مقسوماً على «4». مثلاً:

يستخدم الرمز «S» كذلك لتكبير الرسم، ويمكن لنا ملاحظة ذلك في البرنامج (A62) التالي:

```
5    REM **** A62     *****
10   REM *** DRAW IRREGULAR SHAPE
20   REM *** SET HI-RES MODE
30   SCREEN 2
40   REM *** DRAW DOUBLE SCALE
50   DRAW «S8C6BM100, 90E10R20F10 R40E20R10D30L40E10R20»
60   GOTO 60
```

الفصل التاسع

الأشكال المتحركة

سنتعلم في هذا الفصل كيف نرسم الأشكال بواسطة الكومبيوتر، وكيف نجعل هذه الأشكال تتحرك. وهذا هو مبدأ جميع الألعاب المسلية التي نمارسها مع الكومبيوتر.

الأشكال أو الصور المتحركة هي أساساً صور متتابعة ترسم بسرعة بحيث لا يمكن للعين متابعة سرعة رسمها فتبدو كأنها صورة واحدة متحركة. وبصورة أدق لا يمكن للدماغ تفسير صور متتابعة تختلف عن بعضها البعض قليلاً ويفسرها وكأنها صورة واحدة يتغير شكلها وموقعها بصورة مستمرة. هذه هي نظرية الحصول على الصور أو الأشكال المتحركة، ولنحاول الآن أن نجرب ذلك عملياً. المثال (B1) أدناه هو برنامج لرسم دوائر، الواحدة بعد الأخرى، وبمسافات قليلة جداً، وهذا يعطي لنا انطباعاً بأن الدائرة تتحرك على الشاشة، يحدد السطر 80 المتغيرات الخاصة بمركز الدائرة، وكذلك القطر واللون. ويحدد الدوار (FOR-NEXT) بين السطرين 100 و150، حركة مركز الدائرة، حيث نرسم في السطر 110 دائرة، ونضيف في السطر 130 على المتغير X مقداراً معيناً وبذلك نحرك مركز الدائرة قليلاً نحو جهة اليمين، ثم يعود البرنامج ليرسم دائرة أخرى لها مركز في الموقع الجديد، وهذا يستمر حتى نهاية الشاشة. أما السطر 160 فيبقي الشكل عل الشاشة.

```
5   REM *** B1    ****
10  REM *** X, Y = CIRCLE CENTRE
20  REM *** C = COLOUR
30  REM *** R = RADIUS
40  REM *** DRAW MOVING CIRCLES
50  REM *** SET HI-RES MODE
60  SCREEN 2
```

```
70   REM *** SET PARAMETERS
80   X = 0: Y = 95: C = 1: R = 20
100  FOR LP = 1 TO 50
110  CIRCLE (X,Y), R, C
120  REM *** SET NEW POSITION
130  X = X + 5
140  REM *** START AGAIN
150  NEXT LP
160  GOTO 160
```

الآن حاول أن تغير السطر 130 ليصبح كالآتي:

```
130  X = X + 2
```

وبذلك سوف نلاحظ بأن الحركة تكون أدق حيث ترسم الدوائر بصورة متقاربة لبعضها البعض أكثر. حاول الآن تغيير السطر 130 لتحصل على أبعاد مختلفة للدوائر، وسوف تلاحظ أنه كلما كانت الدوائر متقاربة كانت الحركة أكثر دقة. وعملياً فإن التنسيق بين دقة الحركة والتقارب بين الأشكال مطلوب.

في البرنامج (B1) السابق لاحظنا أن الدوائر التي رسمت تبقى على الشاشة، ولغرض الحصول على حركة مترابطة يجب أن نمسح الدائرة التي على جهة اليسار حال رسم الدائرة الجديدة. ولغرض مسح أي شكل نرسم الشكل نفسه وفي الموقع نفسه ولكن بلون الخلفية للشاشة. البرنامج (B2) التالي يوضح ذلك حيث نحدد في السطر 80 مركز الدائرة، قطرها، واللون الذي ترسم به. واللون الذي ستمسح به الدائرة. السطر 90 يحدد موقع مركز الدائرة الماسحة ويرسم السطر 120 الدائرة. أما السطر 130 فليس له تأثير في البداية، والسطر 150 ينسخ قيمة X لتحدد قيمة A وذلك لرسم دائرة المسح، ولتغيير موقع مركز الدائرة الجديدة. ويعود البرنامج مرة أخرى ليرسم دائرة جديدة في الموقع الجديد. فيرسم السطر 130 الآن دائرة جديدة وبلون الخلفية وفوق الدائرة الأولى وبذلك تمسح الدائرة الأولى. بذلك نلاحظ أن الدائرة تحركت نحو اليمين بموقع واحد. وتستمر هذه العملية حتى انتهاء الدوار «FOR-NEXT» بين السطرين 110 و 170.

```
5     REM *** B2 ****
10    REM *** X, Y = CIRCLE CENTRE
20    REM *** C = COLOUR
30    REM *** R = RADIUS: E = BACKGROUND COLOUR
40    REM *** DRAW MOVING CIRCLE
50    REM *** SET HI-RES MODE
60    SCREEN 2
70    REM *** SET PARAMETERS
80    X = 0: Y = 95: C = 1: R = 20: E = 4
90    A =0: B = 95
100   REM *** DRAW CIRCLE/ERASE CIRCLE
110   FOR LP = 1 TO 255
120   CIRCLE (X,Y), R, C
130   CIRCLE (A,B), R, E
140   REM *** SET NEW POSITION
150   A = X: X = X + 1
160   REM *** START AGAIN
170   NEXT LP
180   GOTO 180
```

ويمكن لنا تحريك أي شكل بالصورة أعلاه. تذكر دائماً أن ترتيب البرنامج يجب أن يكون كالآتي:

- انسخ موقع الرسم لغرض المسح.
- حدد الموقع الجديد للرسم.
- ارسم.
- امسح.
- ابدأ مرة أخرى.

ونستخدم المبدأ نفسه لتحريك مستطيل على الشاشة، نستخدم هنا بعض الخطوات الاضافية لنتحقق فيما إذا وصل المستطيل إلى نهاية الشاشة. وبذلك نعكس اتجاه الحركة. يحدد السطر 90 موقع رسم المستطيل، ويحدد السطر 100 موقع المستطيل الماسح الأول. ويحدد السطر 110 لون المسح. كذلك مقدار التغير في اتجاه (Y, X). والسطر 130 يرسم المستطيل، وأما السطر 140 فليس له تأثير في بادىء الأمر. والسطر 160 يفحص فيما إذا كان المستطيل في أقصى اليسار

أو أقصى اليمين للشاشة. فإذا ما وصل المستطيل إلى أقصى اليمين أو اليسار، فإن إشارة (DX) تتغير. السطر 180 يشابه السطر 160 ويفحص فيما إذا كان المستطيل قد وصل إلى أعلى الشاشة أو إلى أسفلها. فإذا ما وصل المستطيل إلى أعلى الشاشة أو إلى أسفلها فإنه يغير إشارة (DY). وأما السطر (200) فيستنسخ موقع الرسم لغرض المسح، ويحدد السطر 310 الموقع الجديد للرسم تبعاً لقيم وإشارات (DY, DX). (إذا ما كانت «+» فإنها تتحرك باتجاه اليمين أو نحو الأسفل. أما إذا كانت «−» فإنها تتحرك نحو اليسار أو نحو الأعلى). ويعيد السطر 230 الحاسب إلى السطر 130 ليرسم مستطيل في الموقع الجديد. والسطر 140 الآن يرسم المستطيل بلون الخلفية في موقع المستطيل الأول. ويستمر البرنامج لحين الضغط على (CTRL-STOP).

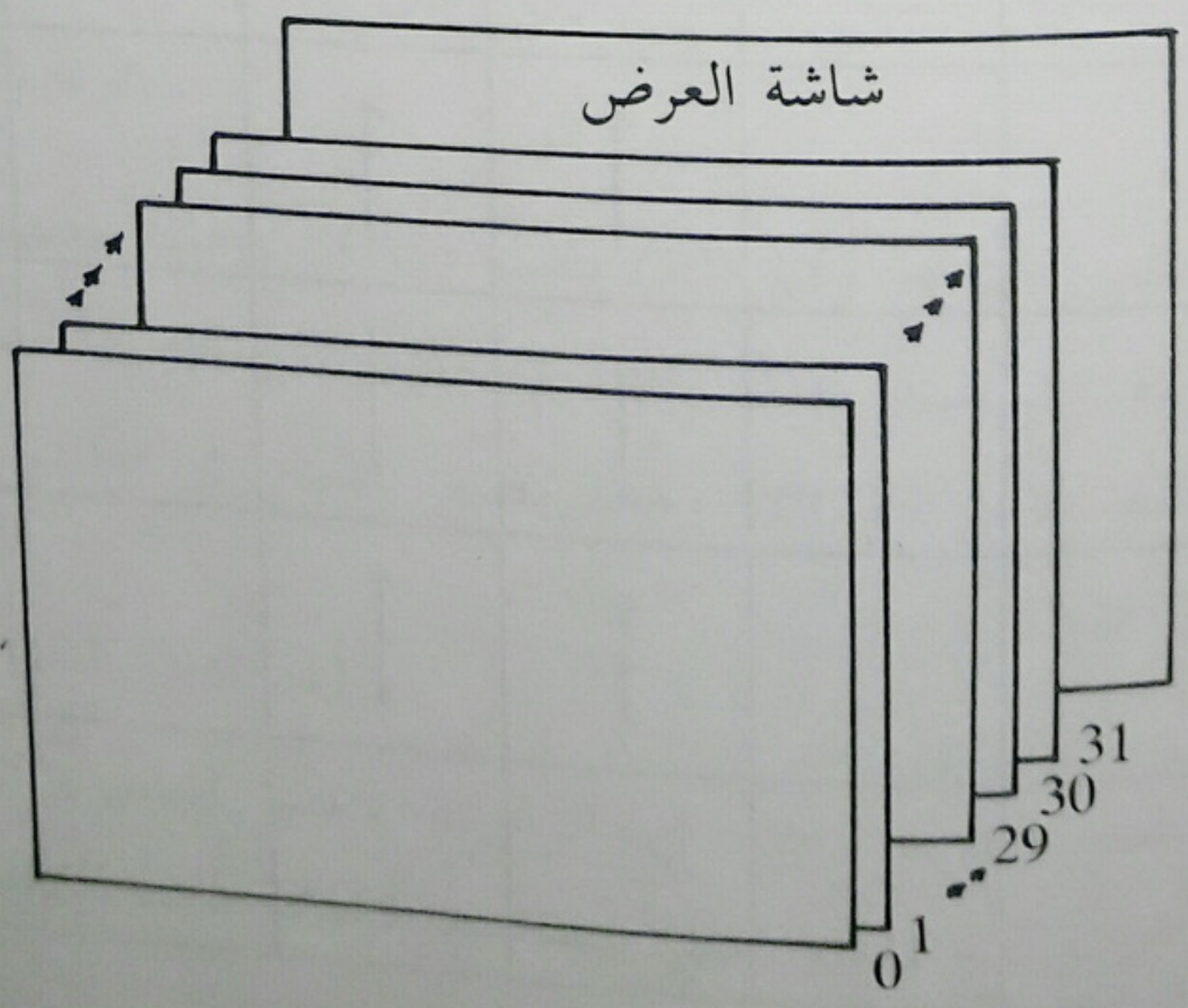
```
5   REM *** B3      ****
10  REM *** E = BLUE DX = X DIRECTION
20  REM *** DY = Y DIRECTION
30  REM *** X, Y, A, B = BOX CO-ORDS
40  REM *** X1, Y1, X1, B1 = ERASE CO-ORDS
50  REM *** DRAW MOVING BOX
60  REM *** SET MULTI COLOUR MODE
70  SCREEN 2
80  REM *** SET PARAMETERS
90  X = 1: Y = 1: A = 21: B = 21
100 X1 = 0: Y1 = 0: A1 = 10: B1 = 10
110 E = 4: DX = 2: DY = 2
120 REM *** DRAW BOX
130 LINE (X, Y) - (A,B), ,B
140 LINE (X1, Y1) - (A1, B1), E, B
150 REM *** CHANGE 1/R DIRECTION
160 IF X > 235 OR X < 1 THEN DX = -DX
170 REM *** CHANGE U/D DIRECTION
180 IF Y > 235 OR Y <1 THEN DY = -DY
190 REM *** SET NEW POSITION
200 X1 = X: Y1 = Y: A1 = A: B1 = B
210 X = X + DX: A = A + DX: Y = Y + DY: B = B + DY
220 REM *** START AGAIN
230 GOTO 130
```

رسم الأشباح : SPRITE, PUT

جميع الأشكال التي رسمناها رسمت في مستوى الرسم (PATTERN PLANE). وجميع الرسوم التي تظهر على الشاشة هي من النوع نفسه حيث تظهر في المستوى نفسه (مستوى الرسم). ولكن هناك 32 مستوى (PLANE) آخر يمكن رسم الأشباح فيها، وتسمى مستويات الأشباح (SPRITE PLANES) وهي مرقمة من (0) إلى (31). ويمكن لكل مستوى من هذه المستويات أن يحتوي عل شبح، والشبح يقصد به هنا شكل مكون من مجموعة من النقاط التي يمكن برمجتها بواسطة الحاسبة وتحريكها على الشاشة بواسطة البرنامج.

جميع المستويات تقع على بعضها البعض، وأولها يكون مستوى الرسم (PATTERN PLANE) حيث يقع في الخلف ومستويات الأشباح تقع أمامه مبتدئة بالمستوى (31)، (أقرب مستوى إلى مستوى الرسم) ومنتهية بالمستوى (0) (أقرب مستوى إلينا).

جميع الأشباح تتحرك في المستويات المحددة لها. هذا يعني إذا ما تحرك الشبح في المستوى «0» وللموقع نفسه لشبح في المستوى «1»، يصبح الشبح «0» فوق الشبخ «1»، ويمنعنا من رؤيته، وهذا صحيح بالنسبة لجميع المستويات الأخرى. وكلما صغر رقم مستوى الشبح أصبح أقرب إلينا وحجب كل ما يقع خلفه عنا.



أربعة أنواع من الأشباح (الصور) يمكن الحصول عليها وهي :

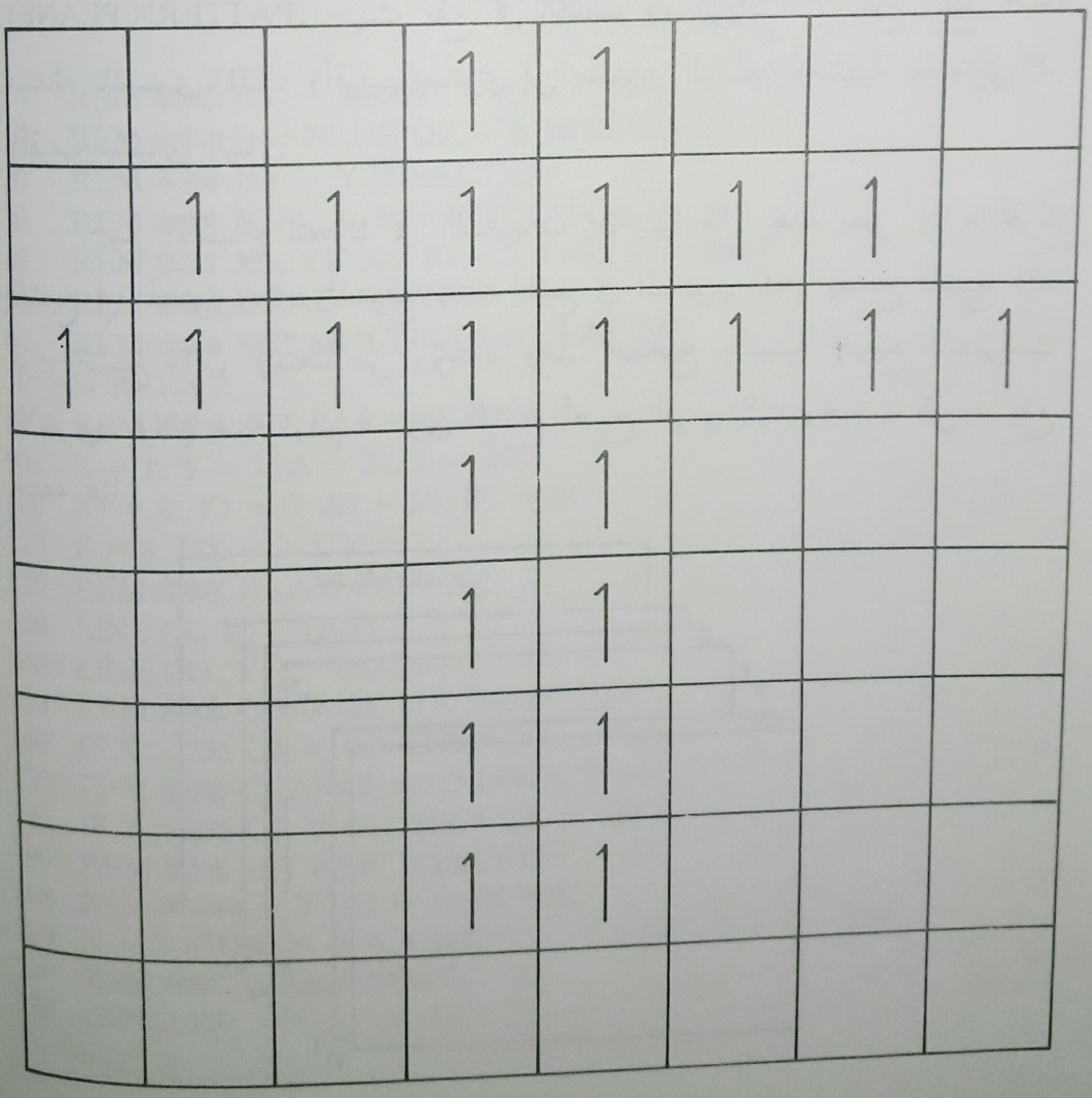
1 ـ 8 × 8 بالحجم الاعتيادي.

2 ـ 8 × 8 بالحجم المضاعف.

3 ـ 16 × 16 بالحجم الاعتيادي.

4 ـ 16 × 16 بالحجم المضاعف.

لغرض الحصول على أشباح ذات (8 ×8) نستخدم مشبكاً ذا (8 × 8) وما نفعله هنا هو تلوين نقاط (فتحات المشبك) للحصول على الشبح المطلوب (كما في الشكل). وأبسط طريقة لعمل ذلك موضحة في البرنامج (B4) القادم.



والفكرة أساساً تعتمد على عملية تمثيل المشبك الثماني باستخدام ثمانية أرقام ثنائية (BINARY). فإذا ما وضعنا «1» في الرقم الثنائي، فإن ذلك يلون النقطة المقابلة لموقعه في الشبح. وجميع النقاط التي تركت «0» تبقى بدون لون. (في الواقع فإنها تكون بلون شفاف)، وعند وضع الأرقام الثنائية الثمانية فإننا نضعها بصيغة حزم (STRINGS) ذات ثمانية أحرف ونمثل الشبح بواسطة هذه الأحرف الثمانية. البرنامج أدناه يوضح ذلك بصورة مفصلة.

نوع الشبح يكتب مع الأمر الخاص بالشاشة، مثلاً:

- يضع هذا البلاغ شبحاً ذا (8 × 8) بالحجم الاعتيادي في النمط (SCREEN 2,0) المتعدد الألوان.
- يضع هذا البلاغ شبحاً ذا (8 × 8) بالحجم المضاعف في النمط (SCREEN 2,1) ذي الدقة العالية (HI-RES).
- يضع هذا البلاغ شبحاً ذا (16 × 16) بالحجم الاعتيادي في النمط (SCREEN 2,2) المتعدد الألوان.
- يضع هذا البلاغ شبحاً ذا (16 × 16) بالحجم المضاعف في النمط (SCREEN 2,3) ذا الدقة العالية (HI-RES).

وهذا ما سوف نلاحظه.

ويمكن تسمية الشبح باسم معين يمكن مناداته به، مثلاً:

```
SPRITE$(10) = A$
```

هذا يسمى الشبح رقم (10) باسم (A$). وحالما نحصل على اسم الشبح يمكن لنا تعيينه وتسميته بذلك الاسم خلال البرنامج دائماً. ولغرض عرض الشبح نستخدم الأمر «PUT» وهذا يضع الشبح في مكان معين على الشاشة وبلون معين. مثلاً:

```
PUT SPRITE 1, (0,0), 11
```

هذا يضع الشبح «1» في الزاوية العليا اليسرى للشاشة وباللون الأصفر (11). البرنامج (B4) التالي يستخدم الشاشة ذات الدقة العالية، والشبح

بالحجم الاعتيادي. وهذا محدد في السطر 130. الأسطر من 150 إلى 220 تحدد شكل الشبح. يضيف السطر 240 الحزم ذات الأحرف الثمانية لنحصل على حزمة واحدة «S1$». والسطر 260 يحدد اسم الشبح. وأما السطر 280 فيضع الشبح على الشاشة في الموقع المحدد له. ستلاحظ في البرنامج أن الرقم «1» في الجمل الثنائية الأرقام هو الذي يحدد شكل الشبح المحدد بالحزم ذات الأحرف الثمانية، وهو التقاطع (الصليب «+»). ويمكن تحديد أي شكل بهذه الطريقة. عندما نريد رسم شبح ما من الأفضل كتابة أسطر البرنامج أولاً. وعرض مكان الشكل على هيئة مربعات فارغة (BINARY BLOCK) على الشاشة وباستخدام الأسهم يمكن تحريك الـ (CURSOR) في المربعات، وبذلك نرسم الشكل الذي نريده، في الأسطر التي تحدد الشكل. ثم نضغط على مفتاح الإدخال (CR)، إدخال الأسطر ضمن البرنامج. ويمكن كذلك رسم الشبح على ورقةمربعات ذات (8 × 8) مربعات.

```
10   REM ***      B4      ***
100  REM *** SPRITE PRODUCTION
110  REM *** HI-RES MODE
120  REM *** 8 * 8 SINGLE SIZE SPRITE
130  SCREEN 2, 0
140  REM *** SET UP SPRITE
150  A$ = CHR$(&B00011000)
160  B$ = CHR$(&B00011000)
170  C$ = CHR$(&B00011000)
180  D$ = CHR$(&B11111111)
190  E$ = CHR$(&B11111111)
200  F$ = CHR$(&B00011000)
210  G$ = CHR$(&B00011000)
220  H$ = CHR$(&B00011000)
230  REM *** TOTAL STRINGS
240  S1$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$
250  REM *** SET SPRITE NUMBER
260  SPRITE $(1) = S1$
270  REM ***  PUT SPRITE ON SCREEN
280  PUT SPRITE 1, (128, 95)
290  GOTO 290
```

يرسم البرنامج (B5) أدناه الشبح السابق نفسه، لكن بالحجم المضاعف. حيث حدد السطر 130 الحجم المضاعف والشاشة ذات الدقة العالية. وحددت الأسطر من 150 إلى 220 شكل الشبح، والسطر 260 رقم الشبح. أما السطر 280 فهو المسؤول عن وضع الشبح على الشاشة وفي الموقع المحدد له.

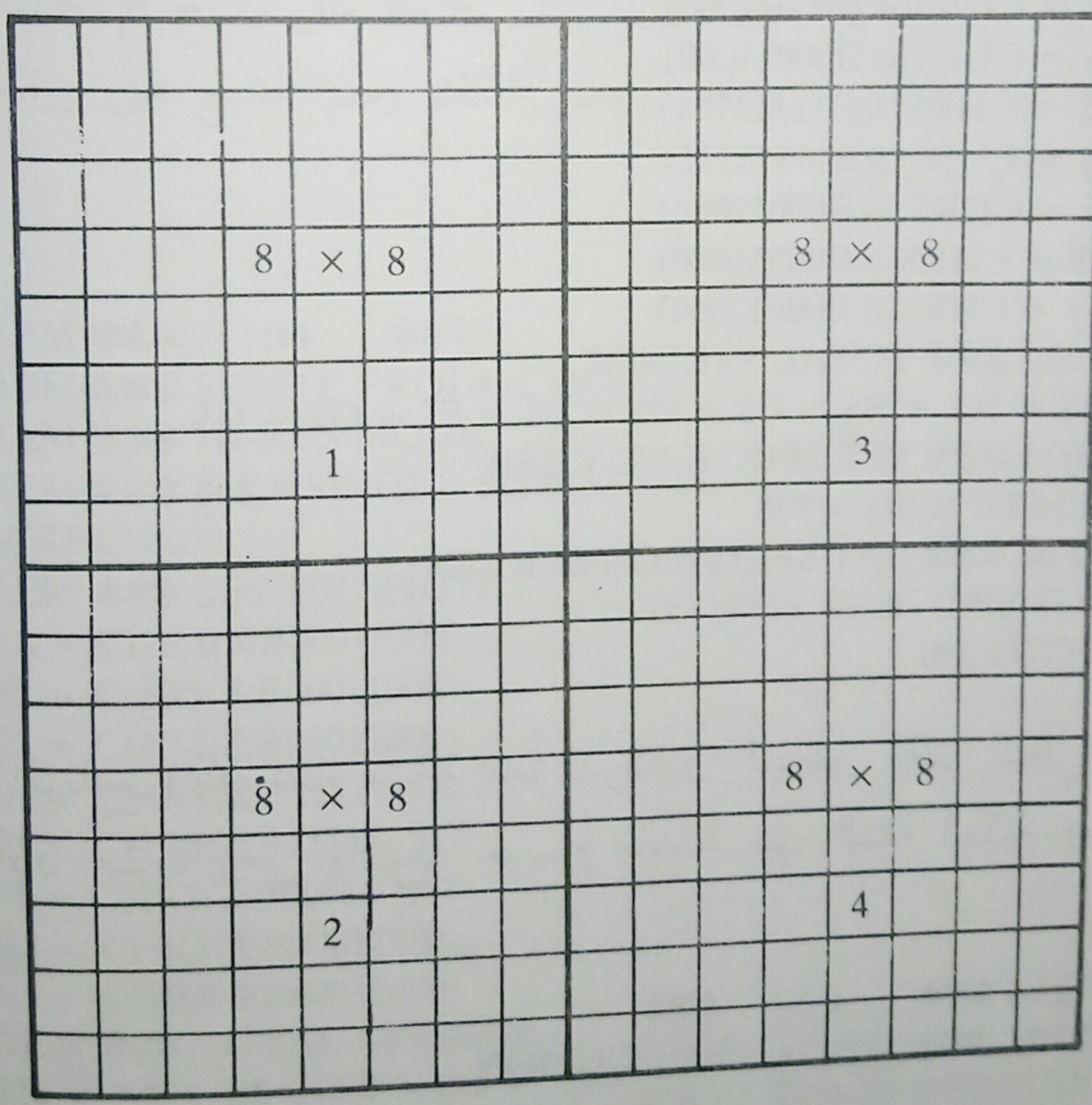
```
10  REM ***     B5     ***
100  REM *** SPRITE PRODUCTION
110  REM *** HI-RES MODE
120  REM *** 8 * 8 DOUBLE SIZE SPRITE
130  SCREEN 2, 1
140  REM *** SET UP SPRITE
150  A$ = CHR$(&B00011000)
160  B$ = CHR$(&B00011000)
170  C$ = CHR$(&B00011000)
180  D$ = CHR$(&B11111111)
190  E$ = CHR$(&B11111111)
200  F$ = CHR$ (&B00011000)
210  G$ = CHR$(&B00011000)
220  H$ = CHR$(&B00011000)
230  REM *** TOTAL STRINGS
240  S1$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$
250  REM *** SET SPRITE NUMBER
260  SPRITE $(10) = S1$
270  REM *** PUT SPRITE ON SCREEN
280  PUT SPRITE 10, (128, 95)
290  GOTO 290
```

البرنامج (B6) أدناه مشابه تماماً للبرنامجين السابقين ولكن شكل الشبح يمثل الآن صورة رجل والشكل مرسوم بالنمط ذي الدقة العالية وبالحجم الطبيعي.

```
10    REM ***     B6     ***
100  REM *** SPRITE PRODUCTION
110  REM *** HI-RES MODE
120  REM *** 8 * 8 SINGLE SIZE SPRITE
130  SCREEN 2, 0
140  REM *** SET UP SPRITE
150  A$ = CHR$(&B00011000)
160  B$ = CHR$(&B00011000)
```

```
170  C$ = CHR$(&B00001001)
180  D$ = CHR$(&B01111110)
190  E$ = CHR$(&B01011100)
200  F$ = CHR$(&B00010100)
210  G$ = CHR$(&B00100100)
220  H$ = CHR$(&B00110110)
230  REM *** TOTAL STRINGS
240  P$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$
250  REM *** SET SPRITE NUMBER
260  SPRITE$(1) = P$
270  REM *** PUT SPRITE ON SCREEN
280  PUT SPRITE 1, (100,25)
290  GOTO 290
```


16



الشبح ذو الـ (16 × 16) ما هو إلا أربعة أشباح ذات الـ (8 × 8) مضافة بعضها لبعض. الأول يحدد موقعه كما سبق والثاني يكون أسفله، والثالث يكون على يمينه، والرابع يكون أسفل الثالث مباشرة. وللسهولة يمكن تصور الشبح

بأنه مكون من نصفين على جهة اليسار ونصفين على جهة اليمين. وباستخدام الأرقام الثنائية يمكن اعتباره حزمة ذات 16 حرف لكل جزء، وهذه الأحرف يجب أن تضاف لبعضها البعض للحصول على حزمة واحدة، ومن ثم نحدد شكل الشبح كما سبق.

البرنامج (B7) القادم يمثل شبحاً ذا (16 × 16) بالحجم الاعتيادي. السطر 100 يحدد النمط ذا الدقة العالية، والشبح ذا الـ (16 × 16) بالحجم الاعتيادي، الأسطر من 120 إلى 270 تمثل الأحرف التي تحدد الجزء الأيمن من الشبح والسطر 290 تضيف الأحرف الممثلة للجزء الأيمن من الشبح للحصول على هذا الجزء كاملاً. الأسطر من 300 إلى 460 تحدد الجزء الأيسر من الشبح. والسطر 500 يضيف جزئي الشبح الأيمن والأيسر لبعضهما للحصول على شكل الشبح. السطر 520 يضع الشبح على الشاشة، والسطر 530 يحافظ على وجود الشبح على الشاشة لكي نتمكن من رؤيته.

```
10   REM *** B7 ***
50   REM *** SPRITE PRODUCTION
60   REM *** HI-RES MODE
70   REM *** 16 * 16 SINGLE SIZE SPRITE
80   REM *** SR$ = SPRITE RIGHT
90   REM *** SL$ = SPRITE LEFT
100  SCREEN 2,2
110  REM *** SET UP RIGHT SIDE
120  A$ = CHR$(&B00000110)
130  B$ = CHR$(&B10001001)
140  C$ = CHR$(&B01110001)
150  D$ = CHR$(&B00000001)
160  E$ = CHR$(&B11110010)
170  F$ = CHR$(&B11111100)
180  G$ = CHR$(&B11111100)
190  H$ = CHR$(&B10011000)
200  I$ = CHR$(&B00011000)
210  J$ = CHR$(&B00110000)
220  K$ = CHR$(&B01100000)
230  L$ = CHR$(&B00000000)
240  M$ = CHR$(&B00000000)
250  N$ = CHR$(&B00000000)
```

```
260  O$ = CHR$(&B00000000)
270  P$ = CHR$(&B00000000)
280  REM *** TOTAL STRINGS
290  SR$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
     + L$ + M$ + N$ + O$ + P$
300  REM *** SET UP LEFT SIDE
310  A$ = CHR$(&B00000000)
320  B$ = CHR$(&B00000000)
330  C$ = CHR$(&B00001000)
340  D$ = CHR$(&B00011100)
350  E$ = CHR$(&B11101111)
360  F$ = CHR$(&B10011111)
370  G$ = CHR$(&B01011111)
380  H$ = CHR$(&B01100011)
390  I$ = CHR$(&B00000011)
400  J$ = CHR$(&B00000110)
410  K$ = CHR$(&B00001100)
420  L$ = CHR$(&B00000000)
430  M$ = CHR$(&B00000000)
440  N$ = CHR$(&B00000000)
450  O$ = CHR$(&B00000000)
460  P$ = CHR$(&B00000000)
470  REM *** TOTAL STRING
480  SL$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
     + L$ + M$ + N$ + O$ + P$
490  REM *** SET SPRITE NUMBER
500  SPRITE $(1) = SL$ + SR$
510  REM *** PUT SPRITE ON SCREEN
520  PUT SPRITE 1, (150, 50)
530  GOTO 530
```

حاول الآن أن تغير السطر 100 ليقرأ كالآتي:

```
100  SCREEN 2, 3
```

هذه سوف تضع الشبح بالحجم المضاعف. لاحظ الفرق في ذلك. لنستخدم الآن الشبح السابق مع تلوينه (باللون الأصفر مثلاً) ونحركه. لغرض تحديد لون معين للشبح نضع رقم اللون المطلوب في نهاية جملة «PUT». ولغرض تحريك الشبح يتطلب الأمر وضع الشبح في موقع مجاور لموقعه الأول. ولا يمكن

وضع الشبح نفسه في موقعين مختلفين على الشاشة في الوقت نفسه، وهذا يختلف عن استخدامنا لمستوى الرسم السابق، حيث كان يتطلب الأمر مسح الشكل السابق. البرنامج القادم (B8) سوف يوضح ذلك.

السطر 60 يضع الشبح في نمط الدقة العالية وبالحجم المضاعف (16 × 16). الأسطر من 70 إلى 460 تضع الشبح بالطريقة السابقة نفسها. ويستمر البرنامج مستخدماً دوار «FOR-NEXT» من السطر 480 إلى 500. السطر 480 يضع الشبح على الشاشة وبموقع يحدد باستخدام المتغير «X». وبتغيير مقدار «MV» من (255) إلى (0) في الدوار «FOR-NEXT» يتحرك الشبح من اليمين نحو اليسار.

```
5    REM *** B8 ***
10   REM *** MOVING SPRITE PRODUCTION
20   REM *** HI-RES MODE
30   REM *** 16 * 16 SINGLE SIZE SPRITE
40   REM *** SR$ = SPRITE RIGHT
50   REM *** SL$ = SPRITE LEFT
60   SCREEN 2, 3
70   REM *** SET UP RIGHT SIDE
80   A$ = CHR$(&B00000110)
90   B$ = CHR$(&B10001001)
100  C$ = CHR$(&B01110001)
110  D$ = CHR$(&B00000001)
120  E$ = CHR$(&B11110010)
130  F$ = CHR$(&B11111100)
140  G$ = CHR$(&B11111100)
150  H$ = CHR$(&B10011000)
160  I$ = CHR$(&B00011000)
170  J$ = CHR$(&B00110000)
180  K$ = CHR$(&B01100000)
190  L$ = CHR$(&B00000000)
200  M$ = CHR$(&B00000000)
210  N$ = CHR$(&B00000000)
220  O$ = CHR$(&B00000000)
230  P$ = CHR$(&B00000000)
240  REM *** TOTAL STRINGS
250  SR$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H $ + I$ + J$ + K$
     + L$ + M$ + N$ + O$ + P$
```

```
260  REM *** SET UP LEFT SIL
270  A$ = CHR$(&B00000000)
280  B$ = CHR$(&B00000000)
290  C$ = CHR$(&B00001000)
300  D$ = CHR$(&B00011100)
310  E$ = CHR$(&B11101111)
320  F$ = CHR$(&B10011111)
330  G$ = CHR$(&B01011111)
340  H$ = CHR$(&B01100011)
350  I$ = CHR$(&B00000011)
360  J$ = CHR$(&B00000110)
370  K$ = CHR$(&B00001100)
380  L$ = CHR$(&B00000000)
390  M$ = CHR$(&B00000000)
400  N$ = CHR$(&B00000000)
410  I$ = CHR$(&B00000011)
420  P$ = CHR$(&B00000000)
430  REM *** TOTAL STRINGS
440  SL$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
     + L$ + M$ + N$ + O$ + P$
450  REM *** SET SPRITE NUMBER
460  SPRITE $(1) = SL$ + SR$
470  REM *** PUT SPRITE ON SCREEN
480  FOR MV = 255 TO 0 STEP  -.5
490  PUT SPRITE 1,(MV,50),11
500  NEXT MV
510  GOTO 510
520  PUT SPRITE 1, (150, 50)
530  GOTO 530
```

إذا ما استخدمنا البلاغ «STEP» في اسطر 480 نستطيع بواسطته أن نحدد سرعة حركة الشبح على الشاشة، حاول ذلك وستجد أنه باستخدام رقم أكبر يمكن أن نحرك الشبح أسرع وتصبح حركته أكثر سلاسة، وعملياً فإن التوافق بين السرعة واتصال الحركة واستمراريتها مطلوب. ولغرض تحريك الشبح نحو الأعلى أو الأسفل يجب تغيير قيمة «Y» بدلاً من «X». حاول أن تضع المتغير «MV» مكان «Y» بدلاً من «X»، في جملة وضع الشبح (PUT STATEMENT) لتحريك الشبح نحو الأعلى أو الأسفل، ويجب التحكم بدوار الـ «FOR-NEXT» بحيث لا يخرج الشبح خارج الشاشة (تذكر الشبكة 191 × 256). حاول تغيير

مقدار كل من «X» و«Y» مع البلاغ «PUT». راجع برنامج المربع المتحرك السابق لتغيير قيم «X» و«Y» وطبق القواعد نفسها.

الصور المتحركة

تعلمنا كيف نرسم الأشباح، وكيف نحركها على الشاشة. والذي سنتعلمه هو الحركة المؤثرة الفعالة للشبح. ما سنقوم به الآن هو وضع شيء من الحياة في الشبح، وهذه ما تعرف بعملية الصور المتحركة. لنأخذ شبح الأسد الصغير ونحاول تحريكه. نستطيع أن نقوم بذلك بكل سهولة برسم شبحين: الأول للأسد الأصلي (الأول)، والثاني مشابه له وفي موقع آخر. نعرض الأول على الشاشة وثم نعرض الثاني في الموقع الآخر، وفي الوقت نفسه نمسح الأول، ثم نعيد عرضه في موقع جديد اخر، وهكذا. هذا يعطي حركة للشبح وحركة فيه.

مسح الشبح يتم بإبدال لونه إلى اللون الشفاف (COLOR,0)، حيث يعطي ذلك في نهاية عبارة الوضع «PUT STATEMENT» بدلاً من لون الشبح المثبت سابقاً وبذلك نكون قد وضعنا الشبح بلون لا يمكن لنا مشاهدته. البرنامج التالي (B9) يوضح ذلك. الأسطر من 100 إلى 480 ترسم الشبح الأول والأسطر من 500 إلى 870 ترسم الشبح الثاني، السطر 890 يضع المتغير «SA» (نستخدم ذلك لظهور واختفاء الشبح). الأسطر بين 900 و990 تمثل دوار «FOR-NEXT». وهناك برنامجان فرعيان (SUBROUTINE) بين السطرين 1010 و1040، الأول لإظهار الشبح الأول ووضع الثاني باللون الشفاف. والثاني لإظهار الشبح الثاني ووضع الأول باللون الشفاف. الأسطر 930 و940 تحدد أياً من البرنامجين الثانويين يختار. وهذه تأتي بأحدهما فقط في كل دورة والآخر في الدورة التي تليها وهكذا، حيث السطر 910 يقلب المتغير «SA» في كل مرة.

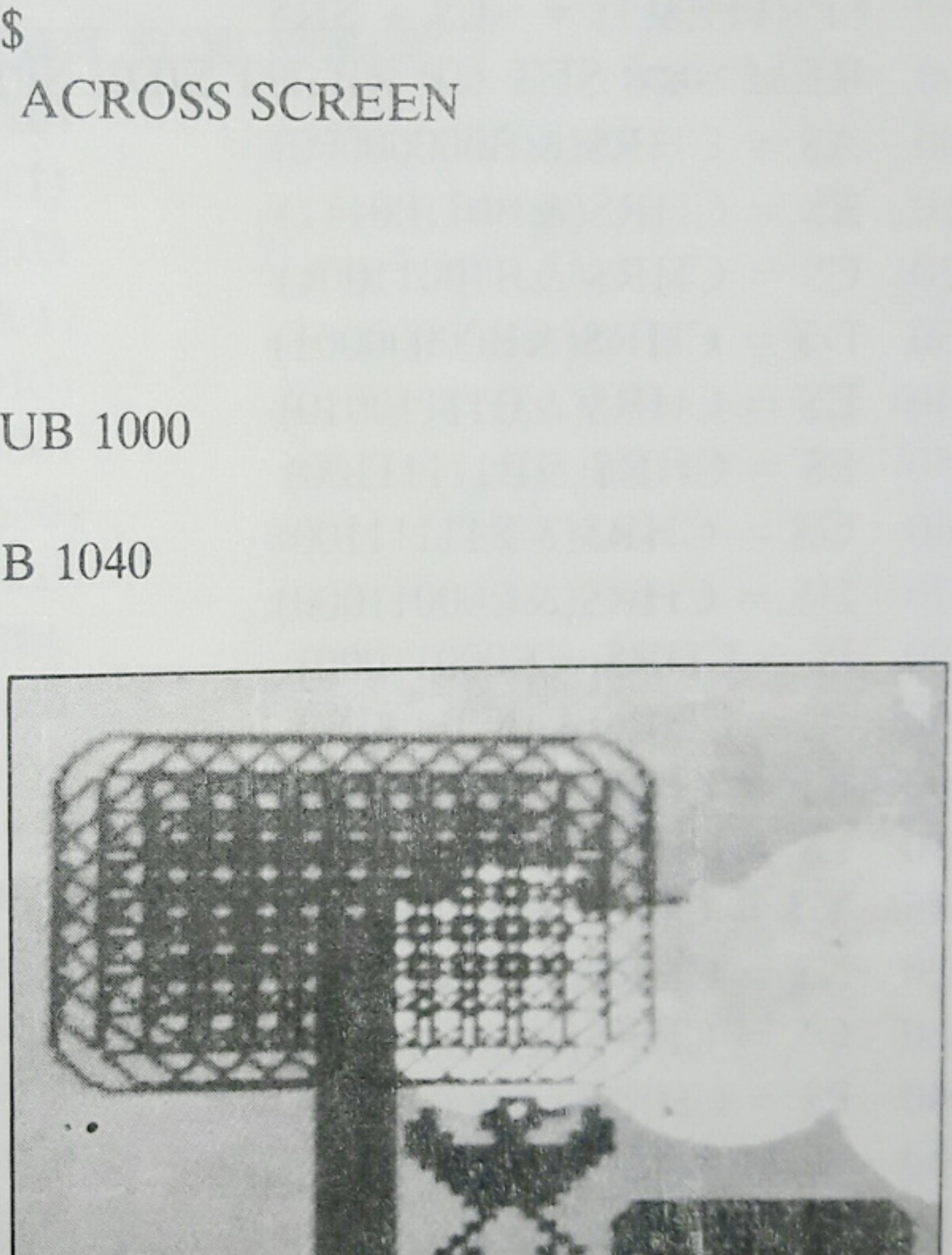
السطر 970 يمثل دواراً يقوم بتأخير بقاء الشبح على الشاشة. يتغير موقع الشبح باستخدام الدوار «FOR-NEXT» في السطر 900 وسوف نلاحظ أن مقدار المتغير «MV» في عبارة الوضع (PUT STATEMENT) يزداد بمقدار «1» في كل

دورة. هذا يعني أن موقع الشبح يكون أعلى بمقدار واحد عن الموقع السابق (الذي أصبح لون الشبح فيه شفافاً).

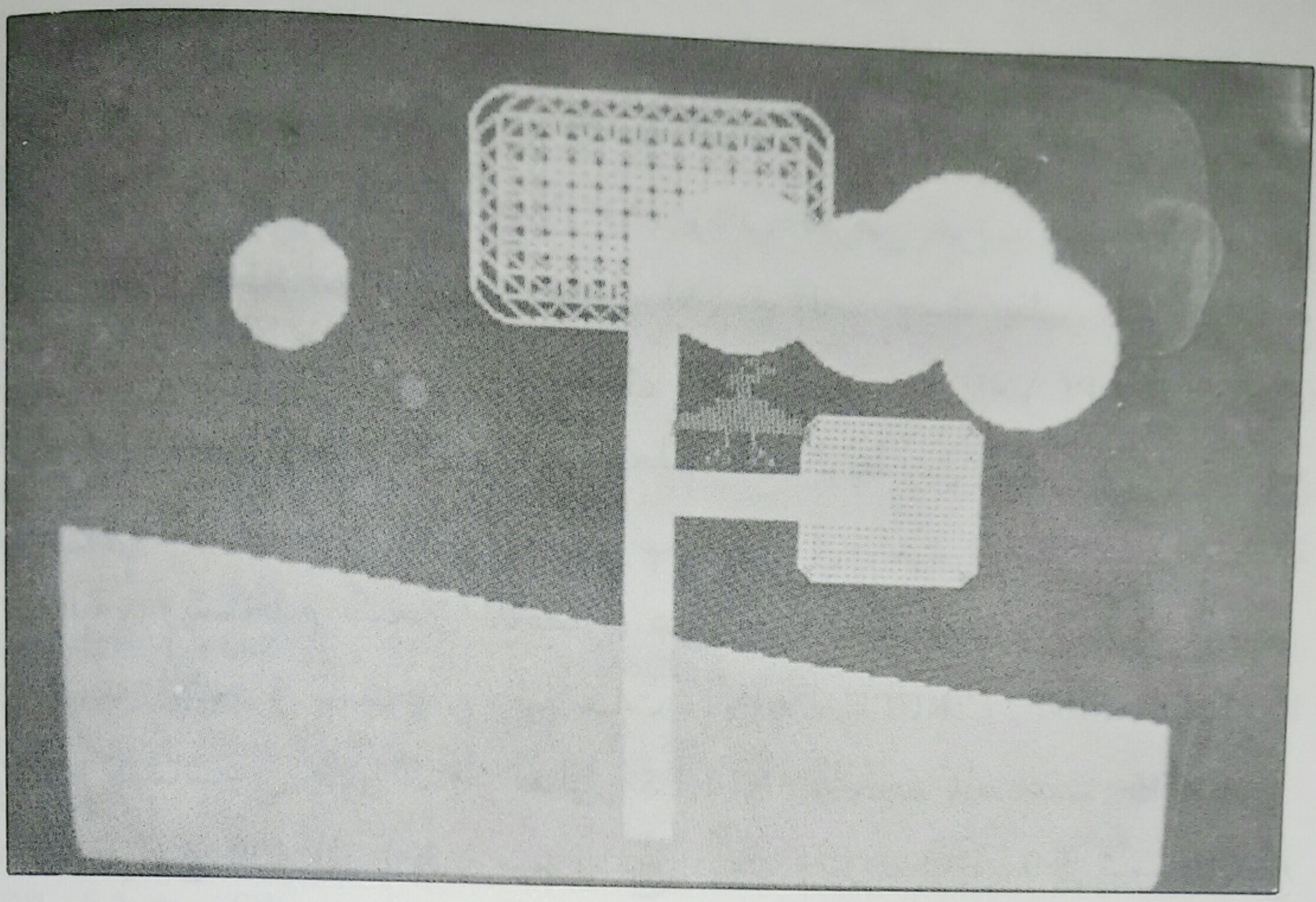
```
5   REM *** B9 ***
10  REM *** ANIMATED SPRITE PRODUCTION
20  REM *** HI-RES MODE
30  REM *** 16 * 16 DOUBLE SIZE SPRITE
40  REM *** SR$ = SPRITE RIGHT
50  REM *** SL$ = SPRITE LEFT
60  REM *** DL = DELAY
70  REM *** SA = SPRITE 1 OR 2 SELECT
80  SCREEN 2, 3
90  REM *** SET UP RIGHT SIDE SPRITE 1
100 A$ = CHR$(&B00000110)
110 B$ = CHR$(&B10001001)
120 C$ = CHR$(&B01110001)
130 D$ = CHR$(&B00000001)
140 E$ = CHR$(&B11110010)
150 F$ = CHR$(&B11111100)
160 G$ = CHR$(&B11111110)
170 H$ = CHR$(&B00011000)
180 I$ = CHR$(&B00011000)
190 J$ = CHR$(&B00011000)
200 K$ = CHR$(&B00110000)
210 L$ = CHR$(&B00000000)
220 M$ = CHR$(&B00000000)
230 N$ = CHR$(&B00000000)
240 O$ = CHR$(&B00000000)
250 P$ = CHR$(&B00000000)
260 REM *** TOTAL STRINGS
270 SR$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
    + L$ + M$ + N$ + O$ + P$
280 REM *** SET UP LEFT SIDE SPRITE 1
290 A$ = CHR$(&B00000000)
300 B$ = CHR$(&B00000000)
310 C$ = CHR$(&B00001000)
320 D$ = CHR$(&B00011100)
330 E$ = CHR$(&B11101111)
340 F$ = CHR$(&B10011111)
350 G$ = CHR$(&B01011111)
360 H$ = CHR$(&B01100111)
```

```
370   I$ = CHR$(&B00000110)
380   J$ = CHR$(&B00001100)
390   K$ = CHR$(&B00011000)
400   L$ = CHR$(&B00000000)
410   M$ = CHR$(&B00000000)
420   N$ = CHR$(&B00000000)
430   O$ = CHR$(&B00000000)
440   P$ = CHR$(&B00000000)
450   REM *** TOTAL STRINGS
460   SL$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
      + L$ + M$ + N$ + O$ + P$
470   REM *** SET SPRITE NUMBER
480   SPRITE$(1) = SL$ + SR$
490   REM *** SET UP RIGHT SIDE SPRITE 2
500   A$ = CHR$(&B00000000)
510   B$ = CHR$(&B01100111)
520   C$ = CHR$(&B10011001)
530   D$ = CHR$(&B00000001)
540   E$ = CHR$(&B11110010)
550   F$ = CHR$(&B11111100)
560   G$ = CHR$(&B11111100)
570   H$ = CHR$(&B00011000)
580   I$ = CHR$(&B00011000)
590   J$ = CHR$(&B00110000)
600   K$ = CHR$(&B01100000)
610   L$ = CHR$(&B00000000)
615   M$ = CHR$(&B00000000)
620   N$ = CHR$(&B00000000)
630   O$ = CHR$(&B00000000)
640   P$ = CHR$(&B00000000)
650   REM *** TOTAL STRINGS
660   SR$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
      + L$ + M$ + N$ + O$ + P$
670   REM *** SET UP LEFT SIDE SPRITE 2
680   A$ = CHR$(&B00000000)
690   B$ = CHR$(&B00000000)
700   C$ = CHR$(&B00001000)
710   D$ = CHR$(&B00011100)
720   E$ = CHR$(&B11101111)
730   F$ = CHR$(&B10011111)
740   G$ = CHR$(&B01011111)
750   H$ = CHR$(&B01100111)
```

```
760  I$ = CHR$(&B00000110)
770  J$ = CHR$(&B00000011)
780  K$ = CHR$(&B00000110)
790  L$ = CHR$(&B00000000)
800  M$ = CHR$(&B00000000)
810  N$ = CHR$(B00000000)
820  O$ = CHR$ (&B00000000)
830  P$ = CHR$(&B00000000)
840  REM *** TOTAL STRINGS
850  SL$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
     + L$ + M$ + N$ + O$ + P$
860  REM *** SET SPRITE NUMBER
870  SPRITE$(2) = SL$ + SR$
880  REM *** MOVE LION ACROSS SCREEN
890  SA = 1
900  FOR MV = 0 TO 255
910  SA = -SA
920  REM *** SPRITE 1?
930  IF SA = -1 THEN GOSUB 1000
940  REM *** SPRITE 2?
950  IF SA = 1 THEN GOSUB 1040
960  REM *** DELAY
970  FOR DL = 1 TO 30:
     NEXT DL
980  NEXT MV
990  GOTO 990
1000 REM *** DISPLAY
      1 ERASE 2
1010 PUT SPRITE 1, (MV +
      1, 90), 11
1020 PUT SPRITE 2,(MV,
      90), 0
1030 RETURN
1040 REM *** DISPLAY 2
     ERASE 1
1050 PUT SPRITE 2, (MV +
      1, 90), 11
1060 PUT SPRITE 1, (MV,
      90), 0
1070 RETURN
```



في البرنامج السابق (B9) نلاحظ أن الأسد يتحرك من اليسار إلى اليمين على الشاشة، هذا حيث أن مقدار (X) يزداد. لغرض تحريك الشبح من اليمين إلى اليسار يجب أن نقلل من مقدار (X) بـ «1»، حيث يظهر الشبح بالموقع «MV-1». بعد أن تعلمنا الرسم في مستوى الرسم (PATTERN PLANE) وكذلك رسم الأشباح المتحركة، سنتعلم الآن الربط بين الاثنين لكتابة برنامج يحتويهما معاً. البرنامج التالي (B10) يرسم صورة في مستوى الرسم، وفي طور الدقة العالية، ويستخدم شبحين للحصول على الحركة لطير يحط على غصن شجرة. رسمت الشجرة باستخدام البلاغات «LINE, CIRCLE, PAINT» السطر 100 يضع الشاشة في طور الدقة العالية «HI-RES» والحجم المضاعف «16 × 16». الأسطر من 120 إلى 500 ترسم الشبح الأول ومن 520 إلى 900 ترسم الشبح الثاني. السطر 920 يرسم خطاً على الشاشة والسطر 930 يلون الجزء الأسفل من الشاشة (في طور الدقة العالية يكون اللون المستخدم، لون الاطار الخارجي نفسه). الأسطر من 950 إلى 1000 ترسم ثلاث دوائر ذات مواقع متباعدة قليلاً وتلونها باللون الأبيض (لتمثل الغيوم). السطران 1040, 1050 يرسمان الشجرة باستخدام البلاغ «LINE» مع «BF». السطران 1070, 1080 يرسمان قرص الشمس باستخدام البلاغ «CIRCLE» مع نسبة بين القطرين بمقدار «1.2» ليكون الشكل بيضاوياً، كذلك البلاغ «PAINT». الأسطر من 1100 إلى 1130 تستخدم الدوار «FOR..NEXT» لتحديد مواقع دوائر لتمثل أغصان الشجرة في الأعلى. والأسطر 1150 إلى 1180 تستخدم الفكرة نفسها لرسم أغصان الشجرة في الأسفل. البرنامج الرئيسي يتمثل بالدوار «FOR..NEXT» في الأسطر 1200 إلى 1290، حيث استخدم المتغير «MV» للمحور «Y» بالنسبة للشبح. الأسطر 1250 إلى 1290، هي دوار «FOR...NEXT» لتغيير عرض الشبحين مع بعضهما، أي اختيار (SP = 1) في إحدى الدورات و (SP = 2) في الدورة التي تليها. نستخدم السطرين 1230, 1250 لنقل البرنامج إلى أحد البرنامجين الثانويين، السطر 1270 يؤخر بقاء الشبح على الشاشة لإتاحة الفرصة لمتابعته، الأسطر 1320, 1330 والأسطر 1360, 1370 ترسم الشبح في موقعه الجديد وتمسح القديم وذلك بتغيير



لونه إلى اللون الشفاف. اطبع البرنامج (B10) التالي ولاحظ كيف يتحرك الشبح على الشاشة في مستوى الرسم، ولاحظ كذلك كيفية رسم الصور في هذا المستوى كخلفية ومن ثم رسم الشبح بعدها للحيلولة دون التداخل بين الصور والألوان.

```
5     REM *****     B10     ****
10    REM *** ANIMATED SPRITE PRODUCTION ***
20    REM *** OVER PATTERN PLAN ***
30    REM *** PICTURE ***
40    REM ** SR$ = SPRITE RIGHT ***
50    REM ** SL$ = SPRITE LEFT
60    REM ** DELAY
70    REM ** SP = SPRITE 1 OR 2
80    REM ** HI-RES MODE
90    REM *** 16 * 16 DOUBLE SIZE SPRITE
100   SCREEN 2,3
110   REM *** SET UP RIGHT SIDE SPRITE 1
120   A$ = CHR$(&B00000000)
130   B$ = CHR$(&B00000000)
140   C$ = CHR$(&B11000000)
150   D$ =CHR$(&B01110100)
160   E$ = CHR$(&B11000110)
```

```
170   F$ = CHR$(&B10001110)
180   G$ = CHR$(&B11111110)
190   H$ = CHR$(&B11111100)
200   I$ = CHR$(&B11111000)
210   J$ = CHR$(&B11110000)
220   K$ = CHR$(&B11100000)
230   L$ = CHR$(&B01000000)
240   M$ = CHR$(&B00100000)
250   N$ = CHR$(&B00010000)
260   O$ = CHR$(&B00101000)
270   P$ = CHR$(&B00000000)
280   REM *** TOTAL STRINGS
290   SR$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
      + L$ + M$ + N$ + O$ + P$
300   REM *** SET UP LEFT SIDE SPRITE 1
310   A$ = CHR$(&B00000000)
320   B$ = CHR$(&B00000000)
330   C$ = CHR$(&B00000001)
340   D$ = CHR$(&B00100011)
350   E$ = CHR$(&B01100011)
360   F$ = CHR$(&B01110001)
370   G$ = CHR$(&B01111111)
380   H$ = CHR$(&B00111111)
390   I$ = CHR$(&B00011111)
400   J$ = CHR$(&B00001111)
410   K$ = CHR$(&B00000111)
420   L$ = CHR$(&B00000010)
430   M$ = CHR$(&B00000100)
440   N$ = CHR$(&B00001000)
450   O$ = CHR$(&B00010100)
460   P$ = CHR$(&B00000000)
470   REM *** TOTAL STRING
480   SL$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
      + L$ + M$ + N$ + O$ + P$
490   REM *** SET SPRITE NUMBER
500   SPRITE$(1) = SL$ + SR$
510   REM *** SET UP RIGHT SIDE SPRITE 2
520   A$ = CHR$(&B00000000)
530   B$ = CHR$(&B00000000)
540   C$ = CHR$(&B11000000)
550   D$ = CHR$(&B01110000)
560   E$ = CHR$(&B11000000)
```

```
570  F$ = CHR$(&B10000000)
580  G$ = CHR$(&B10000000)
590  H$ = CHR$(&B11100000)
600  I$ = CHR$(&B11111000)
610  J$ = CHR$(&B11111110)
620  K$ = CHR$(&B11111111)
630  L$ = CHR$(&B01000001)
640  M$ = CHR$(&B01000000)
650  N$ = CHR$(&B00100000)
660  O$ = CHR$(&B01010000)
670  P$ = CHR$(&B00000000)
680  SR$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
     + L$ + M$ + N$ + O$ + P$
690  REM *** SET UP LEFT SIDE SPRITE 2
710  A$ = CHR$(&B00000000)
720  B$ = CHR$(&B00000000)
730  C$ = CHR$(&B00000001)
740  D$ = CHR$(&B00000011)
750  E$ = CHR$(&B00000011)
760  F$ = CHR$(&B00000001)
770  G$ = CHR$(&B00000011)
780  H$ = CHR$(&B00000111)
790  I$ = CHR$(&B00011111)
800  J$ = CHR$(&B01111111)
810  K$ = CHR$(&B11111111)
820  L$ = CHR$(&B10000010)
830  M$ = CHR$(&B00000010)
840  N$ = CHR$(&B00000100)
850  O$ = CHR$(&B00001010)
860  P$ = CHR$(&B00000000)
870  REM *** TOTAL STRINGS
880  SL$ = A$ + B$ + C$ + D$ + E$ + F$ + G$ + H$ + I$ + J$ + K$
     + L$ + M$ + N$ + O$ + P$
890  REM *** SET SPRITE NUMBER
900  SPRITE$(2) = SL$ + SR$
910  REM DRAW IN GRASS
920  LINE (-32, 110) - (255, 150), 3
930  PAINT (32, 190), 3
940  REM *** DRAW CLOUDS
950  CIRCLE (150, 40), 20, 15
960  PAINT (150, 40), 15
970  CIRCLE (180, 45), 20, 15
```

```
980  PAINT (180, 45), 15
990  CIRCLE (215, 55), 20, 15
1000 PAINT (215, 55), 15
1010 CIRCLE (200, 35), 20, 15
1020 PAINT (200, 35), 15
1030 REM *** DRAW IN TREE TRUNK
1040 LINE (125, 30) - (135, 180), 14, BF
1050 LINE (135, 100) - (185, 90), 14, BF
1060 REM *** DRAW SUN
1070 CIRCLE (50, 50), 15, 10, , , 1.2
1080 PAINT (50, 50), 10
1090 REM *** DRAW IN TOP BRANCHES
1100 FOR TX = 100 TO 160 STEP 6
1110 FOR TY = 10 TO 50 STEP 6
1120 CIRCLE (TX, TY), 10, 12
1130 NEXT TY, TX
1140 REM *** DRAW IN LOWER BRANCHES
1150 FOR TX = 170 TO 200 STEP 2
1160 FOR TY = 80 TO 110 STEP 2
1170 CIRCLE (TX, TY), 5, 12
1180 NEXT TY, TX
1190 REM *** MOVE BIRD DOWN SCREEN
1200 FOR MV = 32 TO 58
1210 FOR SP = 1 TO 2
1220 REM *** SPRITE 1?
1230 IF SP = 1 THEN GOSUB 1310
1240 REM *** SPRITE 2?
1250 IF SP = 2 THEN GOSUB 1350
1260 REM *** DELAY
1270 FOR DL = 1 TO 40: NEXT DL
1280 NEXT SP
1290 NEXT MV
1300 GOTO 1300
1310 REM *** DISPLAY 1 ERASE 2
1320 PUT SPRITE 1, (135, MV + 1), 8
1330 PUT SPRITE 2, (135, MV), 0
1340 RETURN
1350 REM *** DISPLAY 2 ERASE 1
1360 PUT SPRITE 2, (135, MV + 1), 2
1370 PUT SPRITE 1, (135, MV), 0
1380 RETURN
```

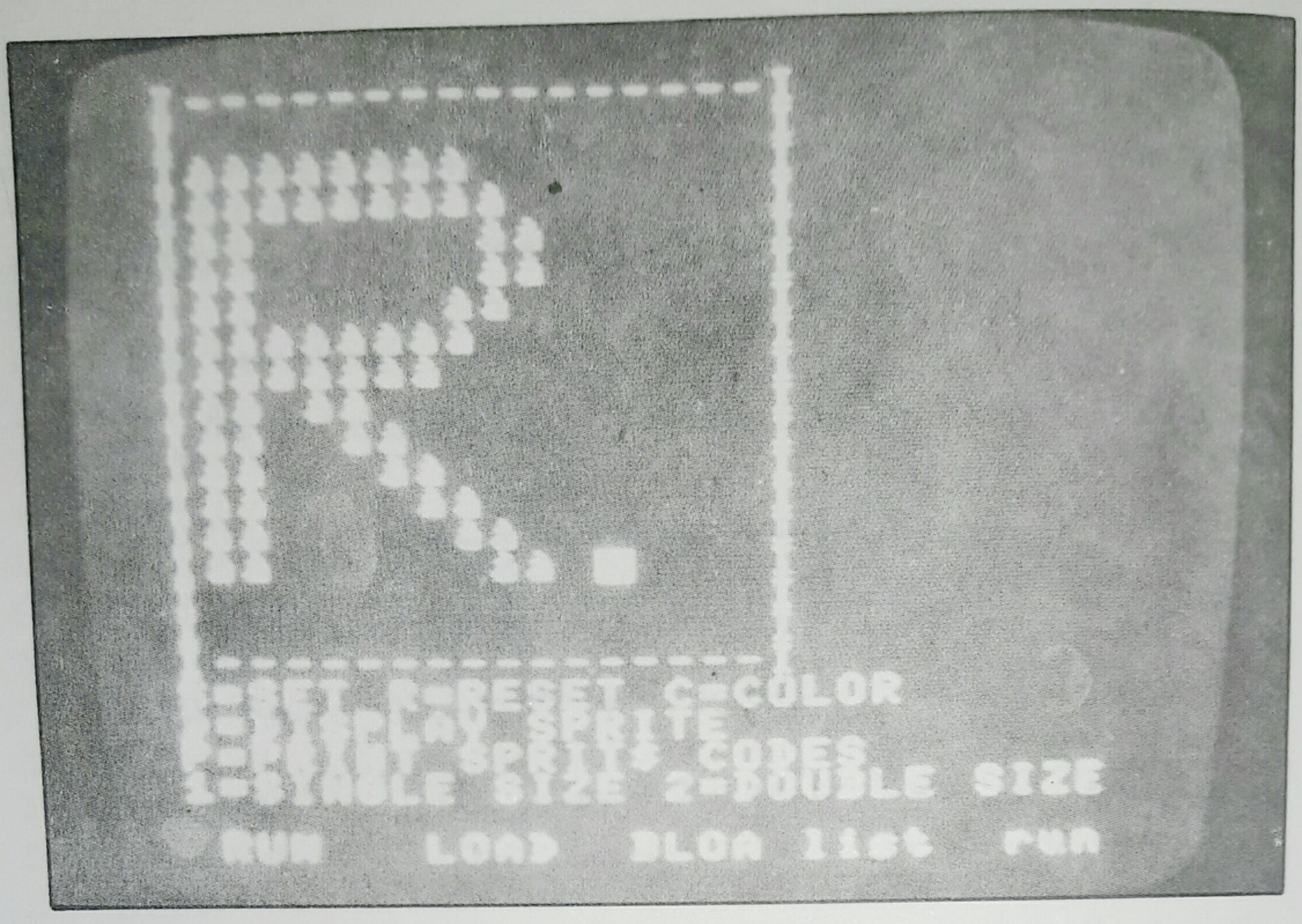
رسم الأشباح باستخدام الحاسب

لاحظنا في البرنامج السابق أن الجزء المستخدم لرسم الشبح كان أكبر من الجزء المستخدم لتحريكه، البرنامج السابق يحتوي على شبحين فقط ومن الممكن استخدام اثنين وثلاثين شبحاً في برنامج واحد وبذلك سوف نستهلك كل الذاكرة قبل إكمال رسم الأشباح، وللحفاظ على الذاكرة، وتسهيل عملية رسم الأشباح، يمكننا استخدام البرنامج (B11) التالي لرسم الأشباح دائماً، ومن ثم عرضها على الشاشة، لذلك يمكن لنا حفظ البرنامج على شريط كاسيت أو قرص، واستخدامه دائماً. هذا البرنامج يستخدم مستقلاً وليس كجزء من برنامج آخر. وبمتابعة البرنامج سنتمكن من فهم الرموز وكيفية استخدامها ضمن البرامج الأخرى. استخدمت الملحوظات «REM» للمساعدة في فهم البرنامج وكذلك في إمكانية تغيره.

```
5   REM **** B11     ***
10  REM *** SPRITE DESIGNER
20  SCREEN 1, 3
30  REM *** CLEAR STRING SPACE
40  CLEAR 2000
50  REM *** DIMENSION ARRAYS
60  DIM GR$(16,16): DIM ST$(16)
70  GX = 1: GY = 1: C = 15
80  LOCATE GX, GY, 1
90  REM *** CLEAR ARRAY
100 FOR X = 1 TO 16
110 FOR Y = 1 TO 16
120 GR$(X,Y) = «   »
130 NEXT Y, X
140 REM *** DRAW SPRITE BOX
150 CLS: LOCATE 0,0,1
160 PRINT «I ---------- I»
170 PRINT «I»;
180 FOR Y = 1 TO 16
190 FOR X = 1 TO 16
200 PRINT GR$(X,Y);
```

```
210  NEXT X: PRINT «I»: PRINT «I»; : NEXT Y
220  REM *** DISPLAY SPRITE
230  PUT SPRITE 1, (200, 50), C
240  LOCATE 0, 17: PRINT «I ---------- I»
250  PRINT «S = SET   R = RESET   C = COLOR»
260  PRINT «D = DISPLAY SPRITE»
270  PRINT «P = PRINT SPRITE CODES»
280  PRINT «1 = SINGLE SIZE    2 = DOUBLE SIZE»
290  REM *** CURSOR MOVEMENT
300  REM *** AND COMMANDS
310  LOCATE GX, GY
320  CM$ = INKEY$
330  IF CM$ = «   » THEN 320
340  CM = ASC(CM$)
350  IF CM = 30 and GY > 1 THEN GY = GY - 1: GOTO 310
360  IF CM = 29 AND GX > 1 THEN GX = GX - 1: GOTO 310
370  IF CM = 28 AND GX < 16 THEN GX = GX + 1: GOTO 310
380  IF CM = 31 AND GY < 16 THEN GY = GY + 1: GOTO 310
390  IF CM$ = «S» THEN GR$(GY, GX) = CHR$(133): PRINT
     CHR$(133): GOTO 310
400  IF CM$ = «R» THEN GR$(GX, GY) = « »:  PRINT «   »: GOTO
     310
410  IF CM$ = «D» THEN GOSUB 490: GOTO 310
420  IF CM$ = «2» THEN SCREEN 1, 3: GOSUB 490: GOTO 150
430  IF CM$ = «1» THEN SCREEN 1, 2: CLS: GOSUB 490: GOTO 150
440  IF CM$ = «C» THEN GOSUB 680: GOTO 230
450  IF CM$ = «P» THEN GOSUB 740: GOTO 150
460  GOTO 310
470  REM ** CONVERT TO SPRITE
480  REM *** CODE AND DISPLAY
490  S1$ = «   »
500  FOR Y = 1 TO 16
510  ST$(Y) = «&B»
520  FOR X = 1 TO 8
530  IF GR$(X,Y) = «   » THEN ST$(Y) = ST$(Y) + «0» ELSE ST$ (Y)
     = ST$(Y) + «1»
540  NEXT X
550  S1$ = S1$ + CHR$(VAL(ST$(Y) ) )
560  NEXT Y
570  FOR Y = 1 TO 16
580  ST$(Y) = «&B»
590  FOR X = 9 TO 16
```

```
600  IF GR$(X,Y) = «   » THEN ST$(Y) = ST$(Y) + «0» ELSE ST$(Y)
     = ST$(Y) + «1»
610  NEXT X
620  S1$ = S1$ + CHR$ (VAL(ST$(Y) ) )
630  NEXT Y
640  SPRITE $(1) = S1$
650  PUT SPRITE 1, (200, 50), C
660  RETURN
670  REM ** GET COLOUR NUMBER
680  C$ = INPUT$(2)
690  C = VAL(C$)
700  IF C < 0 OR C > 15 THEN BEEP: BEEP: BEEP:
     GOTO 680
710  RETURN
720  REM ** GET SPRITE CODES
730  REM ** AND PRINT DECIMAL
740  LOCATE 0, 18
750  PRINT «   »
760  PRINT «   »
770  PRINT «SPRITE$ CODES READ L-R»
780  PRINT «   »
790  FOR Y = 1 TO 16
800  E$ = «&B»
810  FOR X = 1 TO 8
820  IF GR$(X,Y) = «   » THEN E$ = E$ + «0»
     ELSE E$ = E$ + «1»
830  NEXT X
840  PRINT VAL (E$),
850  NEXT Y
860  FOR Y = 1 TO 16
870  E$ = «&B»
880  FOR X = 9 TO 16
890  IF GR$(X,Y) = «   » THEN E$ = E$ + «0»
     ELSE E$ = E$ + «1»
900  NEXT X
910  PRINT VAL (E$),
920  NEXT Y
930  PRINT: PRINT «PRESS ANY KEY TO CONTINUE»
940  IF INKEY$ = «   » THEN 940
950  RETURN
```



المربع الذي يظهر في أعلى الشاشة رسم الحرف «R»

عند تنفيذ البرنامج السابق، يظهر لنا مربع كبير في الجهة اليسرى العليا للشاشة، مساحته «16 × 16» في داخله يمكن لنا رسم الشبح المطلوب وذلك بتحريك المشيرة (CURSOR)، وبالضغط على المفتاح «S» يمكن لنا تثبيت النقطة في مكان وجود المشيرة، وبالضغط عى المفتاح «R» يمكن لنا مسح النقطة. أما تحريك المشيرة فيتم باستخدام مفاتيح الأسهم.

بعد الانتهاء من رسم الشبح في المربع نضغط على المفتاح «D»، وهذا يرسم لنا الشبح على الجهة اليمنى من الشاشة. إذا كان الرسم مقبولاً فهذا شيء جيد، وإذا لم يكن كذلك يمكن استخدام المفتاحين «S» و «R» لتعديل الرسم، والضغط على المفتاح « D » مرة أخرى لنقله إلى الجهة اليمنى من الشاشة. الأمر «D» يأخذ تنفيذه بعض الوقت لذا يجب أن لا نضغط على أي مفتاح إلا بعد إكمال الرسم.

يرسم الشبح بالحجم الاعتيادي، ولرسمه بالحجم المضاعف نضغط على

المفتاح «2»، والضغط على المفتاح «1» يعود بالرسم إلى الحجم الطبيعي إذا ما أردنا أن يكون الرسم بلون محدد، نضغط على المفتاح «C» متبوعاً برمز اللون (يجب أن يكون الرمز مكون من رقمين) مثلاً:

C01 يمثل اللون الأسود

C11 يمثل اللون الأصفر

عند اكتمال رسم الشبح نضغط على المفتاح «P» وبذلك تظهر لنا الرموز الخاصة بالشبح على شكل مجموعتين، وهذه تقرأ من اليسار إلى اليمين. نكتب هذه الرموز على ورقة لكي نستخدمها في برامجنا.

البرنامج التالي (B12) يستخدم مجموعة من الرموز تم الحصول عليها باستخدام برنامج رسم الأشباح (B11) للحصول على شبحين، وضعا على الشاشة في حالة حركة. الأجزاء المهمة في البرنامج هي: رموز الشبحين التي حصلنا عليها باستخدام برنامج رسم الأشباح (B11) ووضعها كبيانات «DATA» في السطرين 320, 350. الأسطر من 60 إلى 90 هي دوار «FOR..NEXT».

يعطي البيانات للشبح الأول، لتكون مجموعة من الرموز (الحزم) لتضاف إلى بعضها البعض لتكون الشبح الأول. الأسطر من 110 إلى 140 تكون الدوار الخاص بالشبح الثاني. الخطين 160, 170 يضعان الأرقام للأشباح بالطريقة الاعتيادية. الأسطر من 190 إلى 250 تعرض الشبحين بالتعاقب. السطران 270, 280 يقومان بتأخير بقاء الشبحين على الشاشة لنتمكن من مشاهدتهما.

```
5     REM **** B12     ****
10    REM ** HOW TO USE
20    REM SPRITE DESIGNER CODES
30    REM SET TEXT SCREEN/SINGLE SIZE SPRITE
40    SCREEN 1, 2
50    REM ** PRODUCE SPRITE 1 STRING
60    FOR DT = 1 TO 32
70    READ SD
```

```
80    S1$ = S1$ + CHR$(SD)
90    NEXT DT
100   REM *** PRODUCE SPRITE 2 STRING
110   FOR DT = 1 TO 32
120   READ SD
130   S2$ = S2$ + CHR$(SD)
140   NEXT DT
150   REM *** DEFINE SPRITES 1 AND 2
160   SPRIT$(1) = S1$
170   SPRITE$(2) = S2$
180   REM *** DISPLAY ALTRNATE SPRITES
190   PUT SPRITE 1, (128, 90), 15
200   PUT SPRITE 2, (128, 90), 0
210   GOSUB 250
220   PUT SPRITE 2, (128, 90), 15
230   PUT SPRITE 1, (128, 90), 0
240   GOSUB 250: GOTO 180
250   BEEP
260   REM *** SOUND AND DELAY
270   FOR DL = 1 TO 100
280   NEXT DL
290   RETURN
300   REM *** SPRITE 1 DATA FROM
310   REM *** SPRITE DESIGNER
320   DATA 31, 32, 64, 131, 133, 128, 176, 208, 192, 192, 179, 129, 128, 64,
      32, 31, 134, 65, 66, 196, 100, 18, 9, 8, 112, 39, 231, 127, 224, 32, 64,
      128,
330   REM *** SPRITE 2 DATA FROM
340   REM **8 SPRITE 2 DESIGNER
350   DATA 31, 32, 64, 131, 133, 128, 176, 208, 192, 192, 179, 129, 128, 64,
      32, 31, 131, 68, 66, 196, 100, 18, 10, 12, 114, 32, 231, 119, 239, 32, 64,
      128
```

تأثير الأشباح على بعضها

وضحنا فيما سبق طريقة رسم وتحريك الأشباح. الموضوع الأخير هو تأثير الأشباح على بعضها، أي حصول حركة في شبح نتيجة حركة شبح آخر، وهذا يمكن ملاحظته في برامج الألعاب، حيث نلاحظ اختفاء الطائرة عند اصطدام القذيفة بها مثلاً، لأجل ذلك نستخدم البلاغات:

ON SPRITE GOSUB عند شبح تفرع

SPRITE ON شبح نعم

SPRITE OFF شبح كلا

هذه البلاغات تستخدم لوضع شبح معين على الشاشة أو لمسحه من عليها إذا ما وضع البلاغ «ON SPRITE GOSUB عند شبح تفرع» ضمن البرنامج ذلك سوف يفحص إذا ما كان هناك شبحان أحدهما فوق الآخر أم لا. الحاسب يقوم بفحص جميع خطوات البرنامج قبل القيام بأي إجراء آخر، لذلك يستخدم هذا البلاغ مرة واحدة فقط ضمن البرنامج، ولأجل أن يكون البلاغ مؤثراً يجب أن نضع الشبح أولاً، أي نستخدم البلاغ «SPRITE ON شبح نعم» وعند الانتهاء نعطي البلاغ «SPRITE OFF شبح كلا» ليصبح البلاغ الأول عديم التأثير. مثلاً:

```
ON SPRITE GOSUB 1000
```

هذا يقود الحاسب إلى البرنامج الفرعي الذي يبدأ بالسطر 1000. ويجب أن يحتوي البرنامج الفرعي على جميع قوانين البرامج الفرعية، حيث يجب أن ينتهي بالأمر «RETURN ارجع» والبرنامج يجب أن لا يدخل ضمن البرنامج الفرعي، حيث عند دخوله البلاغ «SPRITE OFF شبح كلا» يعطيه الحاسب تلقائياً (إحدى خاصيات نظام M.S.X.) لذلك لا حاجة لكتابته داخل البرنامج الثانوي. وعند انتهاء البرنامج الثانوي والوصول إلى الأمر «RETURN ارجع» فإن الحاسب يعطي البلاغ «SPRITE ON شبح نعم» تلقائياً أيضاً.

البرنامج التالي (B13) يمثل لعبة بسيطة، الفكرة أساساً هي عملية رماية أسد وباستخدام البلاغ «ON SPRITE GOSUB عند شبح تفرع». البرنامج يختبر فيما إذا كانت القذيفة قد أصابت الأسد، فإذا ما كانت كذلك فإن الشبح (الأسد) يبدأ بالاختفاء والظهور عدة مرات ثم يعود إلى الجهة اليسرى العليا من الشاشة. في البرنامج استخدم شبحين أحدهما الأسد والآخر هو القذيفة. البيانات الخاصة بالشبحين والتي يمكن الحصول عليها باستخدام برنامج رسم

الأشباح (B11) وضعت في السطرين 500, 520. الأسطر من 80 إلى 200 تقرأ البيانات لترسم الشبحين.

استخدم البلاغين «ON SPRITE GOSUB» و«SPRITE ON» في السطرين 50, 60. واستخدم دوار «FOR..NEXT» لكل من المتغيرين SX, SY واللذين يحددان موقع الشبح (الأسد). واستخدم المتغير (RX) لتغيير موقع القذيفة على المحور (X)، حيث يبدأ مقدار (X) بالنقصان عند الضغط على المفتاح «F». السطران 280, 300 يحددان إحداثيات القذيفة. السطر 320 يفحص فيما إذا وصلت القذيفة إلى نهاية الشاشة العليا أم لا، فإذا ما وصلت فإنها توقف إشارة الرمي وتغير موقع القذيفة. السطر 240 يعرض الشبح الأول (الأسد) والسطر 250 يعرض الشبح الثاني (القذيفة)، ويعمل البلاغ «ON SPRITE GOSUB» عندما يكون الشبحان أحدهما فوق الآخر ويقود الحاسب إلى البرنامج الفرعي حيث أن السطر 400 يرجع القذيفة إلى موقعها الأصلي، والأسطر من 410 إلى 440 تعرض الشبح الأول (الأسد) بصورة متقطعة، والسطر 450 يعيده إلى الزاوية اليسرى العليا للشاشة، السطر 470 يعيد جميع المتغيرات إلى الحالة الابتدائية. السطر 480 يعيد البرنامج مرة أخرى. إذا لم تستطع إصابة الأسد قبل نهاية الدوار «FOR..NEXT» للمتغيرين (SX) و(SY)، فإن السطر 350 يضع الشاشة في نمط الكتابة، والسطر 360 يطبع لنا رسالة. والسطر 370 ينهي البرنامج.

```
10    REM *** B13 ****
20    REM *** INTERRUPT ON SPRITE ***
30    SX = 0: SY = 0 : RX = 160
40    REM *** SET UP INTERRUPT
50    ON SPRITE GOSUB 400
60    SPRITE ON
70    SCREEN 2, 2
80    REM *** READ SPRITE 1 DATA
90    FOR DT = 1 TO 32
100   READ SD
110   S1$ = S1$ + CHR$(SD)
120   NEXT DT
130   REM *** READ SPRITE 2 DATA
```

```
140  FOR DT = 1 TO 32
150  READ SD
160  S2$ = S2$ + CHR$(SD)
170  NEXT DT
180  REM *** DEFINE SPRITES
190  SPRITE$(1) = S1$
200  SPRITE$(2) = S2$
210  REM ** MAIN LOOP
220  FOR SY = 0 TO 150 STEP 20
230  FOR SX = 0 TO 255
240  PUT SPRITE 2, (SX, SY)
250  PUT SPRITE 1, (128, RX)
260  REM *** CHECK FOR FIRE
270  F$ = INKEY$
280  IF F$ = «F» THEN FR$ = «F»
290  REM *** MOVE ROCKET IF FIRE
300  IF FR$ = «F» THEN RX = RX - 10
310  REM *** RESET ROCKET IF OFF SCREEN
320  IF RX < -32 THEN RX = 160: FR$ = «  »
330  NEXT SX, SY
340  REM *** END OF GAME
350  SCREEN 1
360  PRINT «YOU ARE DESTROYED»
370  END
380  REM *** SPRITE CO-INCIDENCE ROUTINE
390  REM *** FLASH  LION  RESET DISPLAY
400  PUT SPRITE 1, (128, 160)
410  FOR BG = 1 TO 100
420  PUT SPRITE 2, (SX + BG, SY), 6
430  PUT SPRITE 2, (SX + BG, SY), 15
440  NEXT BG
450  PUT SPRITE 2, (0,0), 15
460  REM *** RESET VARIABLES
470  SX = 0: SY = 0: RX = 160: FR$ = «  »
480  RETURN
490  REM *** SPRITE 1 DATA
500  DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 3, 5, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 128, 64, 192,
510  REM **** SPRITE 2DATA
520  DATA 15, 15, 9, 9, 31, 31, 27, 51, 99, 99, 102, 102, 192, 192, 0, 0,
     240, 240, 144, 144, 248, 248, 216, 204, 198, 198, 102, 102, 3, 3, 0, 0
```

إذا أردنا أن نزيد من صعوبة إصابة الهدف نبدل السطر 300 ليصبح كالآتي:

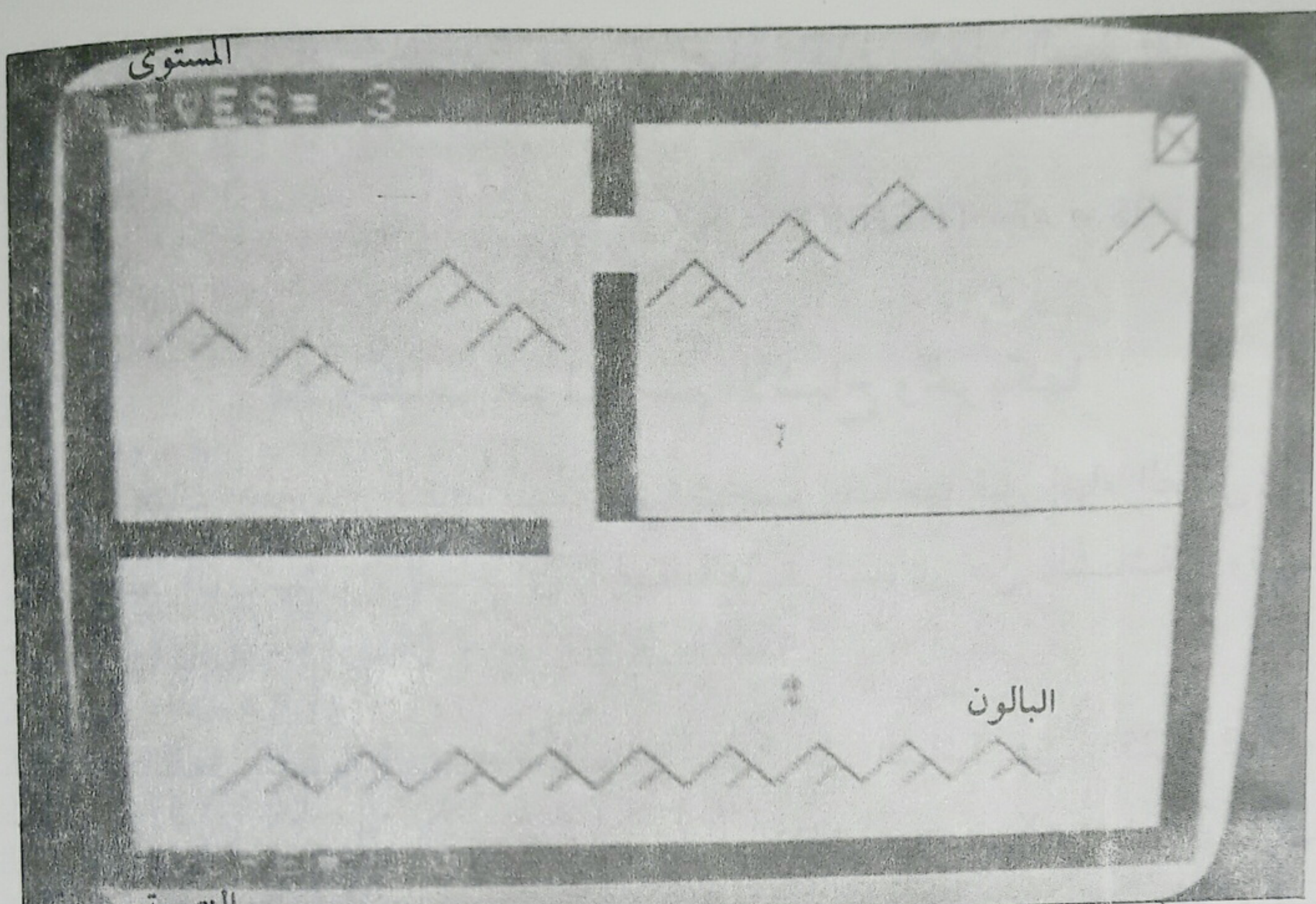
```
300  IF FR$ = «F» THEN RX = RX – 3
```

ملاحظات حول رسم الأشباح وتحريكها

هناك جملة من النقاط المهمة التي يتوجب إيضاحها قبل إنهاء الفصل:

- يجب أن تثبت الأشباح في الجهة العليا اليسرى من الشاشة، وتثبت إحداثياتها.
- إن نقاط الرسم المسموح بها والتي تظهر على الشاشة هي (255 × 192)، ولكن يمكن تحريك الشبح من (32–) إلى (255) للمحور «X» ومن (32–) إلى (191) للمحور «Y»، وهذا يعني إمكانية تحريك الشبح على جزء غير مرئي من الشاشة.
- إشارة بالخطأ سوف تظهر إذا ما وضعت أكثر من أربعة أشباح لها ذات القيمة للمحور «X».

يمثل البرنامج التالي (B14) لعبة (GAME) متكاملة، وتتمثل بوجود بالون يحترق ويهوي نحو الأرض ويتطلب الأمر نقله وإنزاله إلى المنطقة المعلمة بـ «X» في الجزء العلوي الأيمن من الشاشة. ويحترق البالون إذا ما اصطدم بأي شبح عل الشاشة. عند الوصول إلى المربع الحاوي على العلامة «X» تنتقل إلى شاشة أخرى فيها عوائق أكثر من الأولى. وهكذا إلى أن تصل إلى الشاشة الرابعة. ولأجل مشاهدة الشاشات الأربع اضغط على المفتاح «D»، ولبدء البرنامج اضغط على المفتاح «S». يمكنك استخدام مفاتيح الأسهم لتحريك البالون، كذلك يمكنك استخدام عصا التحريك على أن تضغط أولاً على المفتاح «J». خطوات البرنامج واضحة ولا تحتاج إلى أي توضيح إضافي، وهناك خطوط تحمل ملاحظات (REM) لتوضيح تسلسل البرنامج. البرنامج يعتبر مراجعة للأوامر والبلاغات التي درسناها في الفصول السابقة.




```
1     REM *** B14 ***
5     REM *****          BALLOON ***
10    COLOR 1, 15, 15:  KEYOFF: HI = 0:  PLAY «O2G # F #8O3C#
      GD # C # F # D #O2BC # C # F # »: Q= 0: GOSUB 1300: OPEN
      «GRP:» AS # 1: N$ = «M S X»
20    LI = 3: S = 1: CLS: SC = 0
30    SCREEN 2,0
40    RESTORE: A$ = «   »: FOR F = 0 TO 7: READ D$: A$ = A$ +
      CHR$(VAL(«&B» + D$) ): NEXT F: SPRITE$(0) = A$
50    DATA 00001000
60    DATA 00011100
70    DATA 00111110
80    DATA 00111110
90    DATA 00010100
100   DATA 00010100
110   DATA 00011100
120   DATA 00011100
130   REM *** SCREEN ***
140   TIME = 0
150   CLS
160   GOSUB 1910
170   ON S GOSUB 380, 530, 680, 830, 1140
180   DRAW «BM 230, 11C1R10D10L10U10F10L10E10»
```

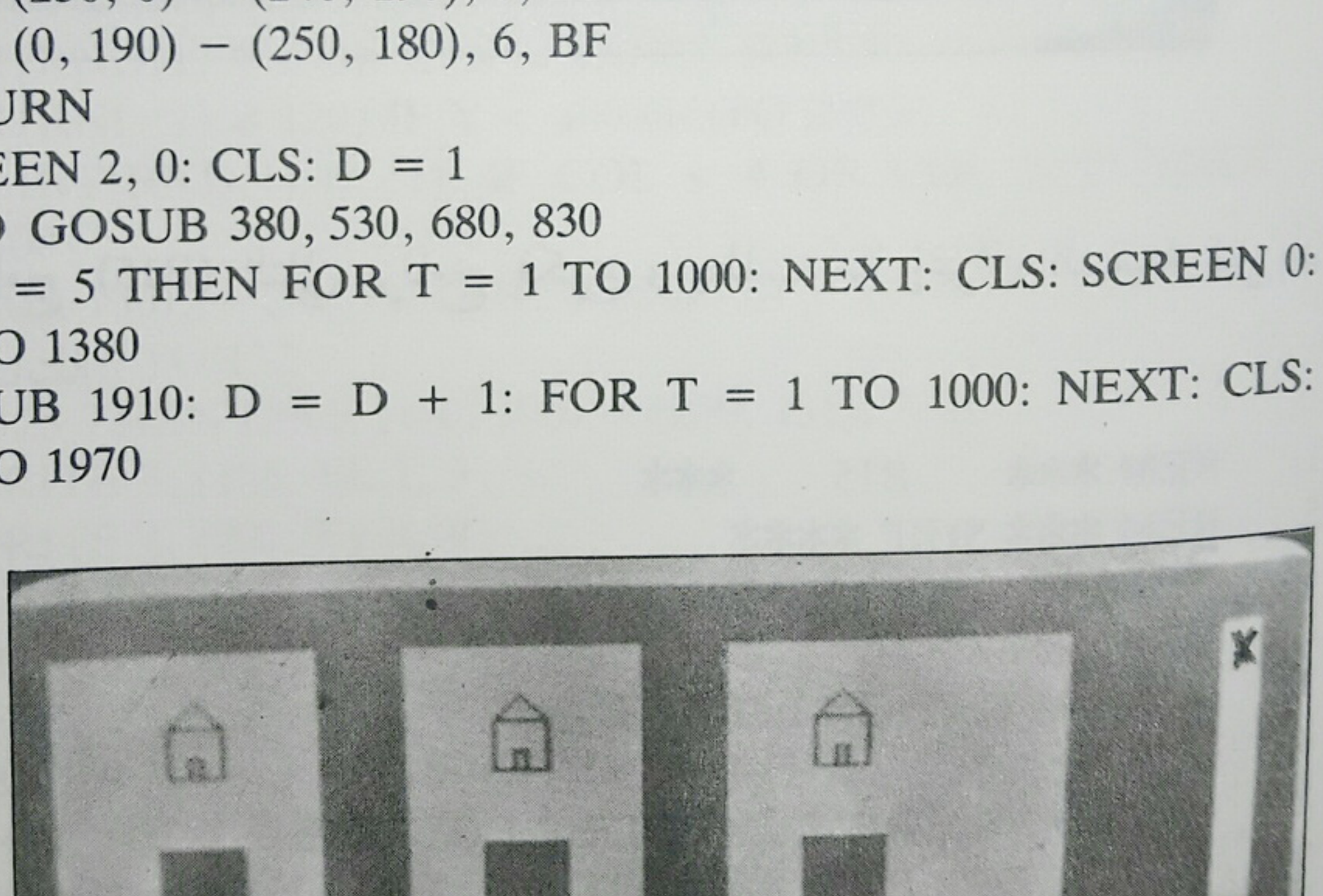
```
190  COLOR 15: PRESET (10, 3): PRINT # 1, «LIVES =»; LI: PRESET
     (10, 180): PRINT #1, «SCORE =    »; SC
200  REM ** MAIN LOOP
210  ST = STICK(Q): IF ST = 0 THEN 250
220  IF ST = 1 THEN Y = Y - 1: GOTO 260
230  IF ST = 3 THEN X = X + 1: GOTO 260
240  IF ST = 7 THEN X = X - 1: GOTO 260
250  Y = Y + 1
260  PUT SPRITE 0, (X, Y), 3
270  IF POINT (X, Y) = 6 THEN GOTO 330
280  IF POINT (X, Y) = 1 THEN GOTO 360
290  X% = X + 7: Y% = Y + 8
300  IF POINT (X%, Y%) = 6 THEN 330
310  IF POINT (X%, Y%) = 1 THEN 360
320  GOTO 210
330  LI = LI - 1: IF LI < 1 THEN 1710
340  PLAY «O4L5CCCCD # DDCCCC»: FOR T = 1 TO 100: NEXT T
350  GOTO 130
360  IF TIME < 3000 THEN SC = SC + 3000 - TIME
370  PLAY «O6DB»: S = S + 1: GOTO 130
380  REM SCREEN 1
390  A$ = «E10F10H5G5»
400  LINE (0, 100) - (100, 108), 6, BF: LINE (111, 100) - (250, 100), 6,
     BF
410  LINE (111, 0) - (119, 30), 6, BF: LINE (111, 43) - (119, 100), 6, BF
420  DRAW «BM84, 60C6XA$;»
430  FOR T = 30 TO 200 STEP 20
440  DRAW «BM = T;, 165C6XA$;»
450  NEXT T
460  DRAW «BM70, 50C6XA$;»
470  DRAW «BM40, 68C6XA$;»
480  DRAW «BM18, 60C6XA$;»
490  DRAW «BM220, 40C6XA$;»: DRAW «BM122, 50C6XA$;»: DRAW
     «BM142, 40C6XA$;»
500  DRAW «BM165, 33CXA$;»
510  X = 150: Y = 130
520  RETURN
530  REM *** SCREEN 2
540  B$ = «R10H5H10D10R4U4R2D4R4U10D10L10»
550  FOR T = 30 TO 230 STEP 11
560  DRAW «BM = T; 50C6XB$;»
570  NEXT
```

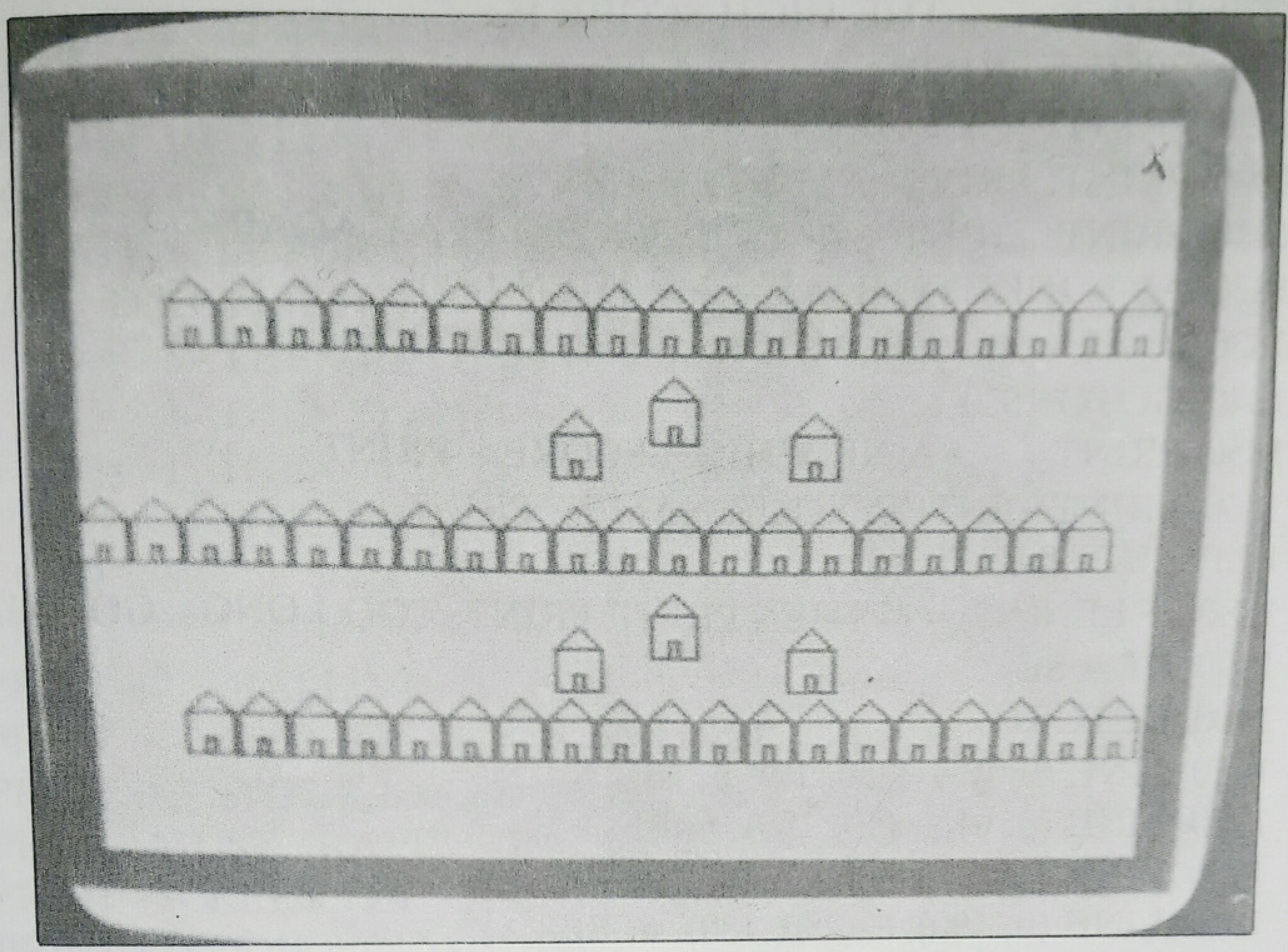
```
580  FOR T = 10 TO 220 STEP 11
590  DRAW «BM = T;  100XB$;»
600  NEXT
610  FOR T = 30 TO 230 STEP 11
620  DRAW «BM = T;  145XB$;»
630  NEXT
640  X = 220: Y = 160
650  DRAW «BM109, 78XB$;»: DRAW «BM130, 70XB$;»: DRAW
     «BM160, 78XB$;»
660  DRAW «BM109, 128XB$;»: DRAW «BM130, 120XB$;»: DRAW
     «BM160, 128XB$;»
670  RETURN
680  REM **** SCREEN 3
690  LINE (30, 50) - (45, 190), 6, BF
700  DRAW «BM32, 25C6XB$;»
710  LINE (60, 0) - (75, 147), 6, BF
720  DRAW «BM62, 165XB$;»
730  LINE (90, 50) - (105, 190), 6, BF
740  DRAW «BM92, 25XB$;»
750  LINE (120, 0) - (135, 147), 6, BF
760  DRAW «BM122, 165XB$;»
770  LINE (150, 50) - (165, 190), 6, BF
780  DRAW «BM152, 25XB$;»
790  LINE (200,0) - (230, 100), 6, BF
800  LINE (200, 112) - (215, 190), 6, BF
810  X = 15: Y = 150
820  RETURN
830  REM *** SCREEN 4
840  LINE (120, 0) - (126, 60), 6, BF
850  X = 200: Y = 160
860  LINE (120, 72) - (126, 135), 6, BF
870  LINE (120, 147) - (126, 190), 6, BF
880  LINE (0, 90) - (70, 96), 6, BF
890  LINE (80, 90) - (250, 95), 6, BF
900  C$ = «U10R12U5R3D15L15»
910  DRAW «BM90, 62XC$;»
920  DRAW «BM97, 35XC$;»
930  DRAW «BM78, 82XC$;»
940  DRAW «BM73, 57XC$;»
950  DRAW «BM200, 140XA$;»
960  DRAW «BM185, 130XA$;»
970  DRAW «BM157, 140XA$;»
```

```
980  DRAW «BM215, 135XA$;»
990  DRAW «BM132, 151XA$;»
1000 FOR T = 20 TO 115 STEP 11
1010 DRAW «BM = T;  156XB$;»
1020 NEXT
1030 DRAW «BM82, 105XB$;»
1040 DRAW «BM60, 105XB$;»
1050 DRAW «BM73, 132XB$;»
1060 D$ = «U30R10D30L10U26BR3R3D3L3U3BD5R3D3L3U3BD5R3
     D3L3U3BD5R3D3L3U3BD5R3D3L3U3»
1070 DRAW «BM220, 78XD$;»
1080 DRAW «BM205, 46XD$;»
1090 DRAW «BM136, 45XD$;»
1100 DRAW «BM136, 87XD$;»
1110 DRAW «BM157, 67XD$;»
1120 DRAW «BM188, 82XD$;»
1130 RETURN
1140 REM END OF GAME GO BACK TO START
1150 SCREEN 0: COLOR 1
1160 CLS
1170 PRINT «WELL DONE BALLOONIST YOU HAVE»
1180 PRINT
1190 PRINT «COMPLETED ALL FOUR SCREENS»
1200 PRINT
1210 PRINT «YOU HAVE SCORED»
1220 PRINT
1230 PRINT «          »; SC
1240 PRINT
1250 PRINT «      PRESS SPACE OR FIRE»
1260 PRINT
1270 PRINT «      FOR THE NEXT JOURNEY»
1280 S = 1: IF STRING (Q)      THEN SCREEN 2, 0: GOTO 40
1290 GOTO 1280
1300 COLOR 1: CLS
1310 PRINT «      BALLOON BURST»: PRINT
1320 PRINT: PRINT: PRINT: PRINT: PRINT
1325 Z$ = CHR$(82) + CHR$(73) + CHR$(89) + CHR$(65) +
     CHR$(68) + CHR$(72) + CHR$(32) + CHR$(77) + CHR$(73) +
     CHR$(82) + CHR$(90) + CHR$(65)
1330 PRINT, Z$
1340 PRINT
1350 K$ = «      'D' TO SEE SCREENS»
```

```
1360 X$ = «BALLOON BURST..... 'S' TO STARŤ..... 'J' TO SET
     JOYSTICK... 'I' FOR INSTRUCTIONS.»
1370 COLOR 1: J$ = K$ + X$
1380 FOR T = 1 TO LEN(J$)
1390 LOCATE 4, 15: PRINT TAB(2); MID$(J$, T, 18)
1400 LOCATE 4, 15: FOR K = 1 TO 100: NEXT
1410  A$ = INKEY$: IF A$ = « » THEN NEXT
1420 IF A$ = «S» THEN RETURN
1430 IF A$ = «J» THEN Q = 1
1440 IF A$ = «I» THEN 1470
1450 IF A$ = «D» THEN 1960
1460 GOTO 1380
1470 CLS
1480 PRINT «      BALLOON BURST»
1490 PRINT
1500 PRINT «     YOUR BALLOON HAS BURST AND YOU»
1510 PRINT «MUST TRAVEL THROUGH FOUR»
1520 PRINT «SCREENS OF TERROR TRYING»
1530 PRINT «TO KEEP IN THE AIR.»
1540 PRINT
1550 PRINT «TO COMPLETE A SCREEN YOU»
1560 PRINT «MUST LAND ON THE BLACK»
1570 PRINT «CROSS IN THE TOP-RIGHT»
1580 PRINT «CORNER.»: PRINT
1590 PRINT «TOCHING ANYTHING ELSE»
1600 PRINT «RESULTS IN THE LOSS OF A»
1610 PRINT «LIFE.»: PRINT
1620 PRINT «BEWARE YOUR BALLOON STARTS»
1630 PRINT «FALLING WHEN THE SCREEN»
1640 PRINT «IS DRAWN.» :PRINT
1650 PRINT «      GOOD LUCK!»
1660 PRINT
1670 PRINT «PRESS SPACE»
1680 A$ = INKEY$: IF A$ = « » THEN 1680
1690 IF A$ >< « »      THEN 1680
1700 CLS: GOTO 1380
1710 COLOR 1: SCREEN 0
1720 CLS: PRINT «      YOU CRASHED BUT YOU SCORED»
1730 PRINT: PRINT
1740 PRINT «      »; SC
1750 IF SC > 1 THEN GOSUB 1850
1760 PRINT: PRINT
```

```
1770 PRINT «      THE HIGH SCORE IS»
1780 PRINT: PRINT
1790 PRINT «      » HI
1800 PRINT: PRINT «      BY: – » Z$      ,
1810 PRINT: PRINT «      PRESS 'S' TO PLAY AGAIN»
1820 A$ = INKEY$: IF A$ = « » THEN 1820
1830 IF A$ = «S» OR A$ = «s» THEN GOTO 20
1840 GOTO 1820
1850 PRINT «      A NEW HIGH SCORE»: PRINT
1860 PRINT «ENTER YOUR NAME»: PRINT
1870 LINE INPUT N$
1880 IF LEN(N$) = 15 THEN PRINT: PRINT «TOO LONG»: GOTO 1260
1890 HI = SC
1900 RETURN
1910 LINE (0,0) – (250, 10), 6, BF
1920 LINE (0, 0) – (10, 190), 6, BF
1930 LINE (250, 0) – (240, 190), 6, BF
1940 LINE (0, 190) – (250, 180), 6, BF
1950 RETURN
1960 SCREEN 2, 0: CLS: D = 1
1970 ON D GOSUB 380, 530, 680, 830
1980 IF D = 5 THEN FOR T = 1 TO 1000: NEXT: CLS: SCREEN 0:
     GOTO 1380
1990 GOSUB 1910: D = D + 1: FOR T = 1 TO 1000: NEXT: CLS:
     GOTO 1970
```





البرنامج (B15) التالي برنامج فكري، يطلب منا إغراق السفينة قبل أن تستطيع إصابتنا.

```
1     REM ***     B15     ***
5     REM *** SHIP ****
10    'SINK THE SHIP BY RIYADH MIRZA
20    'FEB. 1986
30    'AN EDUCATION LOGIC GAME
40    CLEAR 500
50    GOSUB 2070: REM ** SET UP ROUTINE
60    GOSUB 1200: REM ** INPUT LEVEL
70    GOSUB 1470: REM ** GRAPHICS
80    GOSUB 2420
90    PSET (0, 160): COLOR 8: PRINT # 1, «WHAT IS YOUR GUESS»
100   IF INKEY$ < > « » THEN 100
110   T = 150: S = 160
120   FOR F = 0 TO 3
130   A$(F) = INPUT$(1)
140   IF A$(F) = CHR$(13) THEN F = 3
150   IF A$(F) = CHR$(8) THEN A$(F) = « »: LINE (0, 160) - (255,
      168), 7, BF: F = 3: GOTO 90
```

```
160  DRAW «BM» + STR$(T) + «,» + STR$ (S)
170  PRINT # 1, A$(F)
180  T = T + 8: NEXT
190  X$ = A$(0) + A$(1) + A$(2) + A$(3)
200  GU% = VAL (X$)
210  IF GU% < NBR THEN H = 128: V = 117: L = −10: M = 0: N =
     55: P = 5: GOSUB 790
220  IF GU% < NBR THEN PSET (0, 160): PRINT # 1, «SORRY
     THAT WAS TOO LOW!!!»: GOTO 390
230  IF GU% > NBR THEN H = 137: V = 205: L = −14: M = −2: N =
     50: P = 4:  GOSUB 790
240  IF GU% > NBR THEN PSET (0, 160): PRINT # 1, «SORRY
     THAT WAS TOO HIGH!!!!»: GOTO 390
250  IF GR% = NBR THEN H = 137: V = 200: L = −12: M = −5: N =
     5: P = 5: GOSUB 790
260  GOSUB 930
270  FOR F = 1 TO 25
280  X = INT(RND(1) * 250): IF X < 150 GOTO 280
290  Y = INT(RND(1) * 120) IF Y < 100 GOTO 290
300  COL = INT(RND(1) * 11): IF COL < 4 OR COL > 11 THEN
     GOTO 300
310  LINE (200, 130), − (X, Y), COL: NEXT
320  FOR F = 116 TO 132
330  IF F = 125 THEN LINE (150, 100) − (250, 131), 7, BF
340  PUT SPRITE 3, (185, F), 1, 3
350  PUT SPRITE 4, (201, F), 1, 4
360  FOR X = 1 TO 50: NEXT X: NEXT F
370  GOSUB 2420
380  GOTO 1030
390  GOSUB 2420
400  IF Z% = 1 THEN GOTO 500
410  IF X% = 0 THEN GOTO 460
420  H% = L% : L% = L%/2
430  IF L% = UN% THEN CG% = L%: GOTO 510
440  IF L% > UN% THEN CG% = L%: X% = 1: GOTO 510
450  IF L% < UN% THEN CG% = L%: X% = 0: GOTO 510
460  N% = (H% − L%)/2: N% = N% + L%
470  IF N% > UN% THEN H% = N%: GOTO 490
480  L% = N%
490  CG% = N%
500  Z% = 0
510  LINE (0, 160) − (255, 168), 7, BF
```

```
520 PSET (15, 15): PRINT # 1,
    «MY GUESS IS» CG%
530 IF CG% < UN% THEN H=125: V =−80: L = 9: M = −8: N = 8.5:
    P = 110: GOSUB 690
540 IF CG% < UN% THEN PSET (25, 25): PRINT # 1, «OH DEAR
    TOO LOW!!!»: GOSUB 2420
550 IF CG% < UN% THEN GOTO 90
560 IF CG% > UN% THEN H = 86: V = −80: L = 9.5: M = −8.25: N
    = 10: P =110: GOSUB 690
570 IF CG% > UN% THEN PSET (25, 25): PRINT # 1, «OH DEAR
    TOO HIGH!!!»: GOSUB 2420
580 IF CG% > UN% THEN GOTO 90
590 IF CG% = UN% THEN H = 101: V = −69: L = 9.5: M = −8.25: N
    = 10: P = 110: GOSUB 690
600 GOSUB 930
610 FOR F = 1 TO 25
620 X = INT(RND(1) * 100)
630 Y = INT(RND(1) * 100): IF Y < 30 OR Y > 80 GOTO 630
640 COL = INT(RND(1) * 11): IF COL < 4 OR COL > 11 THEN
    GOTO 640
650 LINE (40, 85) − (X, Y), COL: NEXT
660 PSET (25, 25): PRINT # 1, «YOUR NUMBER WAS» CG%:
    GOSUB 2420
670 FLAG = 1: GOTO 1030
680 '
690 ' SHIP TO FORT
700 '
710 SOUND 6, 31: SOUND 8, 16: SOUND 11, 103
720 SOUND 12, 30: SOUND 13, 8: SOUND 7, 55
730 FOR F = 1 TO 400: NEXT F: SOUND 8, 0
740 FOR X = L TO M STEP −1: PSET ((H + (X * N) ), (V + (P + (X
    * X) ) ) ): FOR F = 1 TO 100: NEXT: NEXT
750 FOR F = 1 TO 100: NEXT
760 FOR X = L TO M STEP −1: PRESET ( (H + (X * N) ), (V + (P +
    (X * X) ) ) ). NEXT
770 RETURN
780 '
790 'FORT TO SHIP
800 '
810 SOUND 6, 31: SOUND 3, 16: SOUND 11, 103
820 SOUND 12, 30: SOUND 13, 8: SOUND 7, 55
830 FOR F = 1 TO 400: NEXT F: SOUND 8, 0
```

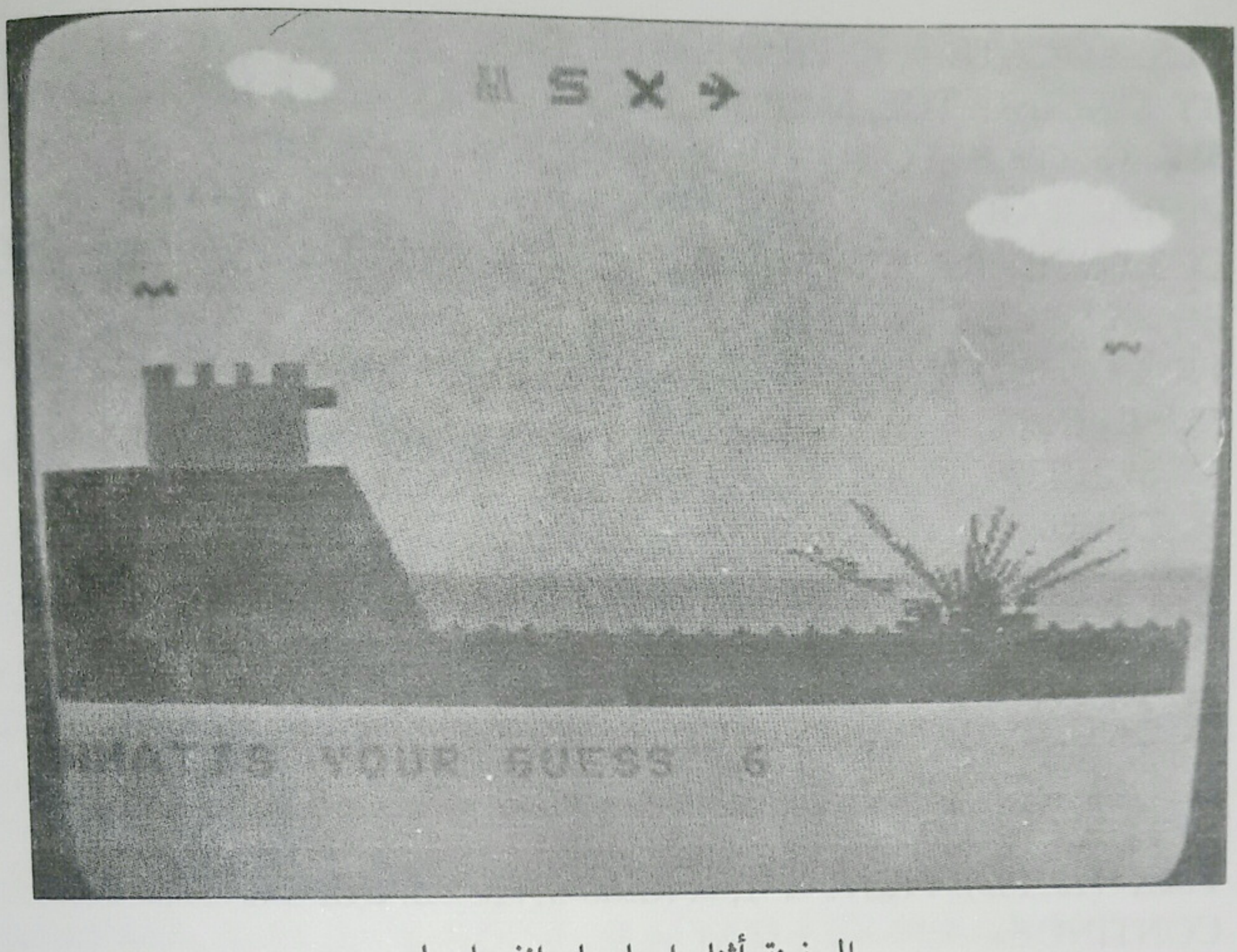
```
840 FOR X = L TO M: PSET (V + (N - (X * X) ), (H + (X * P) ) ):
    FOR F = 1 TO 100: NEXT : NEXT
850 FOR F = 1 TO 100: NEXT
860 FOR X = L TO M: PRESET (V  +  (N - (X * X) ), (H + (X *
    P) ) ): NEXT
870 LINE (0, 155) - (255, 178), 7, BF
880 TRY% = TRY% + 1: PSET (15, 107): PRINT # 1, «TRYS =» TRY%
890 RETURN
900 '
910 'EXPLOSION
920 '
930 SOUND 0, 0: SOUND 1, 5
940 SOUND 2, 0: SOUND 3, 13
950 SOUND 4, 255: SOUND 5, 15
960 SOUND 6, 30: SOUND 7, 0
970 SOUND 8, 16: SOUND 9, 16
980 SOUND 10, 16: SOUND 11, 0
990 SOUND 12, 5: SOUND 13, 0
1000 FOR X = 0 TO 30: NEXT X
1010 SOUND 12, 56: SOUND 13, 0
1020 RETURN
1030 SCREEN 1, 0, 0
1040 SOUND 8, 0: SOUND 9, 0: SOUND 10, 0
1050 IF FLAG = 1 GOTO 1080
1060 PRINT TAB(1); «WELL DONE», NA$
1070 LOCATE 0, 5: PRINT «YOU GUESSED MY NUMBER IN....»
     TRY%; «GUESSES»: GOTO 1100
1080 FLAG = 0: LOCATE 0, 5: PRINT «HOORY, I WON. YOUR
     NUMBER WAS»
1090 LOCATE 12, 8: PRINT UN%
1100 LOCATE 0, 10: PRINT «WOULD YOU LIKE ANOTHER
     GAME?»
1110 CLOSE # 1
1120 LOCATE 8, 22: PRINT «PRESS Y OR N»
1130 IF INKEY$ < > « » THEN 1130
1140 K$ = INKEY$: IF K$ = «Y» OR K$ = «y» THEN GOTO 50 ELSE
     IF K$ = « » THEN 1140
1150 CLS: LOCATE 2, 10: PRINT «THANKS FOR THE GAME.....»
1160 LOCATE 8, 15: PRINT «BYE.. BYE.»: END
1170 '
1180 REM ** INPUT LEVEL
1190 '
```

```
1200 IF INKEY$ < > « » THEN 1200
1210 CLS: LOCATE 0, 6: PRINT «PLEASE TYPE YOUR NAME...»
1220 LOCATE 0, 8: INPUT NA$
1230 LOCATE 0, 10: PRINT «WHICH LEVEL DO YOU REQUIRE»
1240 LOCATE 0, 12: INPUT L: IF L < 1 OR L > 4 THEN CLS:
     GOTO 1230
1250 '
1260 REM *** GET RANDOM NUMBER
1270 '
1280 ON L GOSUB 1350, 1370, 1390, 1410
1290 '
1300 LOCATE 0, 14: PRINT «WHAT NUMBER BETWEEN 1 & 1000
     DO YOU WANT THE COMPUTER TO TRY AND GUESS?»
1310 LOCATE 0, 18: INPUT UN%: IF UN% < 1 OR UN% > 1000
     THEN CLS: GOTO 1300
1320 LOCATE 0, 20: PRINT «THANK YOU» NA$
1330 GOSUB 2370
1340 RETURN
1350 NBR = INT (10 * RND (1) + 1)
1360 RETURN
1370 NBR = INT(100 * RND(1) + 1)
1380 RETURN
1390 NBR = INT(500 * RND(1) + 1)
1400 RETURN
1410 NBR = INT (1000 * RND(1) + 1)
1420 RETURN
1430 '
1440 '
1450 REM *** GRAPHICS
1460 '
1470 SCREEN 2, 2, 0: COLOR 12, 7, 7: CLS
1480 FOR F = 0 TO 255 STEP 8
1490 LINE (F + 4, 132) - (F + 6, 132), 4
1500 PSET (F + 5, 131), 4
1510 NEXT F
1520 LINE (0, 133) - (255, 149), 4, BF
1530 LINE (0, 95) - (63, 95), 12
1540 LINE (63, 95) - (87, 149), 12
1550 LINE (87, 149) - (0, 149), 12
1560 PAINT (0, 148), 12
1570 LINE (23, 71) - (55, 95), 13, BF
1580 FOR F = 29 TO 45 STEP 8
```

```
1590 LINE (F, 71) - (F + 4, 75), 0, BF
1600 NEXT
1610 LINE (56, 77) - (61, 80), 1, BF
1620 CIRCLE (223, 20), 8, 10
1630 PAINT (223, 20), 10
1640 CIRCLE (208, 35), 8, 15, , , .5
1650 CIRCLE (223, 36), 15, 15, , , .5
1660 CIRCLE (235, 38) 11, 15, , .4: LINE (208, 35) - (235, 41), 7:
     PAINT (208, 35), 15, 15
1670 CIRCLE (50, 7), 7, 15, , , .5: CIRCLE (59, 9), 5, 15, , , .5: LINE
     (50, 7) - (59, 7), 7: PAINT (50, 7), 15, 15
1680 RESTORE: B$ = « »: C$ = « »: D$ = « »: E$ = « »: F$ = « »
1690 FOR F = 1 TO 32: READ A: B$ = B$ + CHR$(A): NEXT
1700 FOR F = 1 TO 32: READ A: C$ = C$ + CHR$(A): NEXT
1710 FOR F = 1 TO 32: READ A: D$ = D$ + CHR$(A): NEXT
1720 FOR F = 1 TO 32: READ A: E$ = E$ + CHR$(A): NEXT
1730 SPRITE$(3) = B$
1740 SPRITE$(4) = C$
1750 SPRITE$(1) = D$
1760 SPRITE$(2) = E$
1770 PUT SPRITE 1, (185, 132), 4, 1
1780 PUT SPRITE 2, (201, 132), 4, 2
1790 SOUND 7, 56
1800 PLAY «XT$;», «XT$;», «XT$;»
1810 FOR F = 230 TO 185 STEP -1
1820 PUT SPRITE 3, (F, 116), 1, 3
1830 PUT SPRITE 4, (F + 16, 116), 1, 4
1840 LINE (15, 170) - (255, 178), 7, BF
1850 NEXT F
1860 RESTORE 2510
1870 FOR Y% = 0 TO 5: A$ = « »: FOR X% = 0 TO 7
1880 READ A% : A$ = A$ + CHR$(A%): NEXT X%
1890 SPRITE$(Y% + 25) = A$: NEXT Y%
1900 RETURN
1910 REM *** PLANE/BIRD MOVEMENT
1920 PUT SPRITE 9, (R%, 60), 1, 29
1930 PUT SPRITE 10, (S%, 50), 1, 30
1940 PUT SPRITE 27, (G%, 4), 1, 25
1950 PUT SPRITE 28, (G% - 16, 4), 4, 28
1960 PUT SPRITE 29, (G% - 32, 4), 13, 27
1970 PUT SPRITE 30, (G% - 48, 4), 9, 26
1980 G% = G% + 3
```

```
1990 R% = R% + 1: S% = S% - 1
2000 FOR F = 1 TO 30: NEXT F
2010 PUT SPRITE 9, (R%, 60), 1, 30
2020 PUT SPRITE 10, (S%, 50), 1, 29
2030 FOR F = 1 TO 30: NEXT F
2040 R% = R% + 1: S% = S% - 1
2050 RETURN
2060 '
2070 REM ** SET UP ROUTINE
2080 '
2090 KEYOFF: SCREEN 1, 0, 0: COLOR 1, 2, 2,: CLS: STOP OFF:
     CG% = 1000: L% = 1000: Z% = 1: X% = 1: TRY% = 0: G% = 50:
     R% = 255: S% = 255
2100 T$=«T255CR64CEGR64G2O5GR64G2ER64E2O4CR64CEGR64
     G2O5GR64O2FR64F2O4 DR64 DFAR64 A2O5AR64 A2FR 64F
     2O4DR64DFAR64 A2O5AR 64 A2ER64 EO4CR64 CEGO5C 2O6
     CR64C2O5GR64G2O4CR64CEGO5C2O6CR64C2AR64A2O4D
     DFAR64A. R 64 AF # GO 5 E. R 64 ECO 4 ER 64 E.. R 64 C
     GC.C»
2110 OPEN «GRP:» FOR OUTPUT AS # 1
2120 KEY OFF: LOCATE 8, 10, 0: PRINT «SINK THE SHIP»
2130 LOCATE 0, 20, 0: PRINT «DO YOU REQUIRE
     INSTRUCTIONS?»
2140 LOCATE 8, 22: PRINT «PRESS Y OR N»
2150 IF INKEY$ < > « » THEN 2150
2160 K$ = INKEY$: IF K$ = «Y» OR K$ = «y» THEN GOSUB 2190
     ELSE IF K$ = « » THEN 2160 ELSE CLS: RETURN
2170 RETURN
2190 REM *** INSTRUCTION ROUTINE
2210 CLS: PRINT TAB(9); «INSTRUCTIONS»
2220 PRINT TAB (30); «TO SINK THE SHIP YOU MUST GUESS
     THE NUMBER THAT THE COMPUTER THINKS OF
     BEFORE IT GUESSES YOURS.»
2230 PRINT TAB (29); «JUST TO HELP YOU THE COMPUTER WILL
     GIVE YOU CLUES TO HELP YOU GUESS.»
2240 PRINT TAB (29); «PRESS RETURN AFTER EACH INPUT»
2250 PRINT «YOU HAVE 4 LEVELS AVAILABLE:-»
2260 PRINT TAB (1); «1 A NUMBER BETWEEN 1 & 10»
2270 PRINT TAB (1); «2 A NUMBER BETWEEN 1 & 100»
2280 PRINT TAB (1); «3 A NUMBER BETWEEN 1 & 100»
2290 PRINT TAB (1); «4 A NUMBER BETWEEN 1 & 1000»
2300 GOSUB 2370
```

```
2310 CLS: LOCATE 0, 6: PRINT «THE BEST SCORE IS ACHIEVED
     BY SINKING THE SHIP IN THE        LOWEST NUMBER OF
     SHOTS OR BEFORE YOUR FORT IS BLOWN UP!»
2320 PRINT TAB (29); «YOU WILL BE ABLE TO MAKE THE
     COMPUTER GUESS YOUR INTEGER NUMBER WHICH CAN
     BE BETWEEN 1 & 1000»
2330 PRINT «AND IT WILL USE LOGIC TO WORKOUT WHAT THE
     NUMBER IS...»
2340 GOSUB 2370
2350 RETURN
2360 '
2370 REM ** SPACEBAR ROUTINES
2380 '
2390 LOCATE 1, 23: PRINT «PRESS SPACEBAR TO CONTINUE»
2400 A = STRIG (0): IF A = 0 THEN 2400
2410 RETURN
2420 PSET (15,183): PRINT # 1, «PRESS SPACEBARE TO
     CONTINUE»
2430 A = STRIG(0): GOSUB 1910: IF A = 0 THEN 2430
2440 LINE (10, 183) - (255, 191), 7, BF
2450 LINE (15, 15) - (190, 33), 7, BF
2460 RETURN
2470 DATA 0, 0, 0, 0, 0, 0, 63, 63, 7, 255, 255, 63, 31, 15, 7, 3, 0, 0, 0, 0,
     15, 9, 201, 207, 207, 207, 255, 102, 255, 255, 255, 255
2480 DATA 30, 30, 30, 30, 254, 245, 254, 255, 255, 255, 255, 102, 255, 255,
     255, 255, 0, 0, 0, 15, 15, 15, 8, 8, 255, 255, 255, 127, 254, 252, 252, 248
2490 DATA 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
     255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
     255, 255, 255, 255, 255
2500 DATA 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
     255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
     255, 255, 255, 255, 255
2510 DATA 48, 24, 204, 254, 255, 12, 24, 48, 102, 102, 102, 102, 219, 219,
     219, 219, 127, 255, 192, 254, 127, 3, 255, 254, 193, 227, 119, 62, 28, 62,
     119, 99
2520 DATA 0, 66, 165, 153, 0, 0, 0, 0, 0, 0, 0, 153, 165, 66, 0, 0
```



السفينة أثناء إصابتها وانفجارها

الصوت والموسيقى

يستخدم في الحاسبات المايكروية نظام خاص لتوليد المؤثرات الصوتية ويستخدم هذا النظام مع البرامج، وخصوصاً برامج الألعاب. ويتم التحكم بالإشارات الصوتية من خلال ثلاث قيم أساسية، هي:

1 ـ فترة المكوث: وهي طول الفترة التي تمكث خلالها النغمة الصوتية.
2 ـ التردد: ويحدد نوع الصوت المراد إصداره من حيث الضخامة والحدة.
3 ـ الدرجة: وتحدد مستوى الصوت من حيث درجة الارتفاع والانخفاض.

وتتضمن وسائل التحكم في الصوت في نظام (M.S.X.) الدوال التالية:

*** SOUND *** صوت

يستخدم البلاغ لتوليد مؤثر صوتي وذلك بكتابة البيانات مباشرة على السجالات (REGISTORS) الخاصة بتوليد الصوت داخل الحاسب، ويستخدم البلاغ بالصيغة التالية:

```
SOUND A, B
```

حيث تمثل «A» رقم السجالة و «B» البيان.

نفذ البرنامج (C1) التالي لتأخذ فكرة عن استخدام البلاغ.

```
5     REM *** C1 ***
10    SOUND 0, 87
20    SOUND 7, 62
30    SOUND 8, 16
40    SOUND 11, 179
50    SOUND 12, 45
60    SOUND 13, 14
```

ولكل سجالة وظيفة خاصة لمعالجة الصوت. ويلخص الجدول التالي وظائف هذه السجالات:

رقم السجالة	الدالة	نطاق قيمة البيانات للسجالة
0	تردد القناة A	255-0
1		31-0
2	تردد القناة B	255-0
3		31-0
4	تردد القناة C	255-0
5		31-0
6	تردد الضوضاء	31-0
7	اختيار قناة لنغمة الصوت وقناة توليد الضوضاء	63-0
8	درجة الصوت للقناة A	القيمة 16 تعطي درجة ثابتة للصوت. أما طبقة الصوت فتتم السيطرة عليها بواسطة السجلات 11، 12، 13
9	درجة الصوت للقناة B	
10	درجة الصوت للقناة C	
11	تردد نمط تغيير الدرجة	255-0
12	الصوت	255-0
13	اختيار نمط تغيير درجة الصوت	15-0

البرنامج C2 التالي يعطي فكرة عن عمل كل سجالة. نفذ البرنامج وستظهر لك أرقام السجالات على الشاشة، وبإمكانك وضع البيانات التي

تحددها في كل سجالة على الشاشة، وبإمكانك وضع البيانات التي تحددها في كل سجالة وسماع الصوت الناتج لتحتار المؤثرات الصوتية المناسبة لبرنامجك.

```
10  '*************************
20  '***  SOUND GENERATOR ***
30  '***  RIYADH MIRZA    ***
40  '***  1986            ***
50  '*************************
80  CLEAR 255
90  REM *** INITIALIZE ***
100 DIM A%(14): DIM DV%(30): DIM RN%(14): DIM SV%(14)
110 FOR N = 0 TO 13: DV%(N) = -1: NEXT
120 SCREEN 1, 1, 0: COLOR 1, 13, 13, KEYOFF: CLS: WIDTH 30
130 GOTO 890
140 REM *** GOSUB TO DEFINE SPRITE ***
150 SCREEN 1, 1, 0: CLS: COLOR 1, 13, 5: WIDTH 30
160 GOSUB 800
170 LOCATE 0, 10: PRINT «MIX»
180 LOCATE 5, 10: PRINT «A      B      C»
190 LOCATE 19, 10: PRINT «VOLUME»: LOCATE 19, 11: PRINT
    «PATTERN»
200 LOCATE 7, 11: PRINT «VOLUME»
210 REM *** FIRST SCREEN DISPLAY ***
220 LOCATE 24, 23: PRINT «WHITE»
230 LOCATE 24, 24: PRINT «NOISE»
240 LOCATE 8, 24: PRINT «CHANNELS»
250 X = 2: Y = 20: FOR M = 1 TO 3
260 READ D$: LOCATE X, Y: PRINT «<»; D$; «>»;
270 X = X + 8: NEXT M
280 DATA «A», «B», «C»
290 Y = 16: YA = 0: M = 14: YB = 24: YC = 4: AX = 10: AY = 133:
    A = 0: SL = 208: BX = 0: PP = 14
300 FOR B = 0 TO 13: READ D: SV% (B) = D: NEXT B
310 DATA 255, 15, 255, 15, 255, 15, 31, 63, 16, 16, 16, 255, 255, 14
320 FOR N = YA TO YB STEP YC
330 LOCATE N, Y, 0: PRINT A;
340 A = A + 1
350 NEXT N
360 IF Y = 16 THEN Y = 5: GOTO 320
```

```
370 REM *** MAIN LOOP FOR DATA READ IN ***
380 IF AX MOD (33) = 10 THEN CL% = 11 ELSE CL% = 4
390 PUT SPRITE 1, (AX, AY), CL%, 1
400 K$ = INKEY$: IF K$ = « » THEN 400
410 K = ASC(K$): IF K = 28 THEN AX = AX + 33
420 IF AX > 208 AND PP = 12 THEN AX = 10: AY = 40: BX = 241:
    M = 3: PP = 3: GOTO 380
430 IF K = 29 THEN AX = AX - 33: IF AX < 10 AND BX = 241
    THEN AX = 208: AY = 135: BX = 0: PP = 14: M = 14
440 IF PP = 3 AND AX > 208 THEN AX = 208
450 IF AX < 10 AND AY = 135 THEN AX = 10
460 XX = AX + BX
470 RN% = INT(XX/33)
480 IF XX = 10 THEN RN% = 0
490 IF CL% = 11 AND K = 30 THEN DV%(RN%) = DV%(RN%) + 1
500 IF CL% = 11 AND K = 31 THEN DV%(RN%) = DV%(RN%) - 1
510 P = INT (AX/8.5)
520 IF RN% < 2 THEN NP = P - 1
530 IF RN% = 7 THEN NP = P - 1
540 IF RN% = 9 THEN NP = P + 1
550 IF K = 32 AND CL% = 11 THEN 560 ELSE 580
560 LOCATE 0, 0, 0: INPUT «NUMBER FOR THIS REGISTER»; B$:
    A1 = VAL(B$)
570 DV%(RN%) = A1
580 IF DV%(RN%) < -1 THEN DV%(RN%) = -1
590 IF RN% > 7 AND RN% < 10 THEN P = P - 1
600 IF RN% /2 = INT(RN%/2) THEN PP = M - 2
610 IF RN% / 2 < > INT (RN% / 2) THEN PP = M
620 IF DV%(RN%) > SV%(RN%) THEN DV%(RN%) = SV%(RN%)
630 IF DV%(RN%) = -1 THEN LOCATE P + 1, PP: PRINT «OFF»:
    GOTO 650
640 LOCATE P, PP: PRINT DV%(RN%); SPC(3)
650 LOCATE P + 1, PP + 1: PRINT «XWY»
660 LOCATE 0, 0: PRINT SPC(29)
670 REM *** INTERRPUT TO UPDATE SOUND REGISTERS ***
680 ON INTERVAL = 5 GOSUB 730
690 INTERVAL ON
700 GOTO 380
710 END
720 REM *** LOOP TO GENERATE SOUND ***
730 FOR N = 0 TO 13
740 IF DV%(N) < -1 THEN DV%(N) = -1 :GOTO 770
```

```
750  IF DV%(N) = -1 THEN GOTO 770
760  SOUND N, DV%(N)
770  NEXT N
780  INTERVAL OFF: RETURN
790  REM *** DEFINE SPRITE ***
800  A% = BASE(7) + ASC(«N») * 8
810  A$ = « »
820  FOR N = 0 TO 7
830  C = VPEEK (A% + N): A$ = A$ + CHR$(C)
840  NEXT N: SPRIT$(1) = A$
850  IF PP = 15 THEN 700
860  A% = BASE(6) + 2: VPOKE A%, &H3D
870  VPOKE BASE(6) + 6, &HFD: VPOKE BASE(6) + 7, &HFD
880  RETURN
890  COLOR 1, 5, 5: CLS
900  A% = BASE(6)
910  VPOKE A% + 6, &H55: VPOKE A% + 4, &H55: VPOKE A% +
     5, &HF5: VPOKE A% +7, &H55
920  LOCATE 5, 2: PRINT «THE INSTRUCTIONS FOR»
930  LOCATE 0, 4: PRINT «SOUND GENERATOR SCROLL IN THE»
940  LOCATE 0, 6: PRINT «SAME WAY AS LISTINGS. USE THE»
950  LOCATE 3, 8: PRINT «STOP BUTTON TO HOLD AND»
960  LOCATE 7, 10: PRINT «RELEASE SCROLL»
970  KEY 1, «GOTO 140» + CHR$(13)
980  LOCATE 3, 16: PRINT «PRESS SPACEBAR TO START»
990  K$ = INKEY$: IF K$ = « » THEN 990
100 IF K$ = « » THEN LIST 1010 —    ELSE GOTO 990
1010 'USE THE LEFT/RIGHT KEYS TO MOVE THE YELLOW
     POINTER TO THE DESIRED REGISTER NUMBER, (BELOW
     THE GREEN LINE.) THEN INCREASE OR DECREASE
     REGISTER VALUE (ABOVE GREEN LINE.) WITH THE
     UP/DOWN ARROW KEYS.
1020 'IF GREATER ALTERNATIONS ARE DESIRED THIS IS DONE
     BY PRESSING ««SPACE BAR»» THE REQUIRED VALUE CAN
     NOW BE ENTERED FOLLOWED BY ««RETURN»». ILLEGAL
     VALUES ARE IGNORED AND THE REGISTER IS THEN SET
     TO ITS HIGHEST «VALUE».
1030 'TO MOVE BETWEEN TOP AND BOTTOM ROW MOVE
     RIGHT 'THROUGH SIX OR LEFT THROUGH SEVEN.
     REGISTERS NOT REQUIRED IN A SOUND SHOULD BE SET
     TO ZERO. TO READ LISTING TYPE «SCREEN 1» FIRST TO
     RESTORE LINE NUMBERS.
```

```
1040 'THE SOUND IS HEARD CONTINUOISLY DURING
     ALTERNATION AND MAY BE HEARD IN SHORT BURSTS BY
     MOVING THE LEFT/RIGHT ARROW KEYS. WAIT FOR A
     MOMENT AFTER CHANGING VALUES TO ALLOW TIME
     FOR REGISTER UPDATE AND FEEDBACK ON THE EFFECT.
1050 'AS ON SCREEN INFORMATION IS LIMITED YOU SHOULD
     CONSUT YOUR USER MANUAL FOR DETAILS OF SOUND
     REGISTER FUNCTIONS. REMEMBER IF ALL VOLUME
     CHANNELS EIGHT TO TEN ARE OFF NO SOUND IS HEARD.
1055 '
1056 '
1060 'PRESS F ONE TO START
```

*** BEEP

*** بيب

يستخدم البلاغ «BEEP» كما هو لإصدار صوت شبيه بـ «بيب».

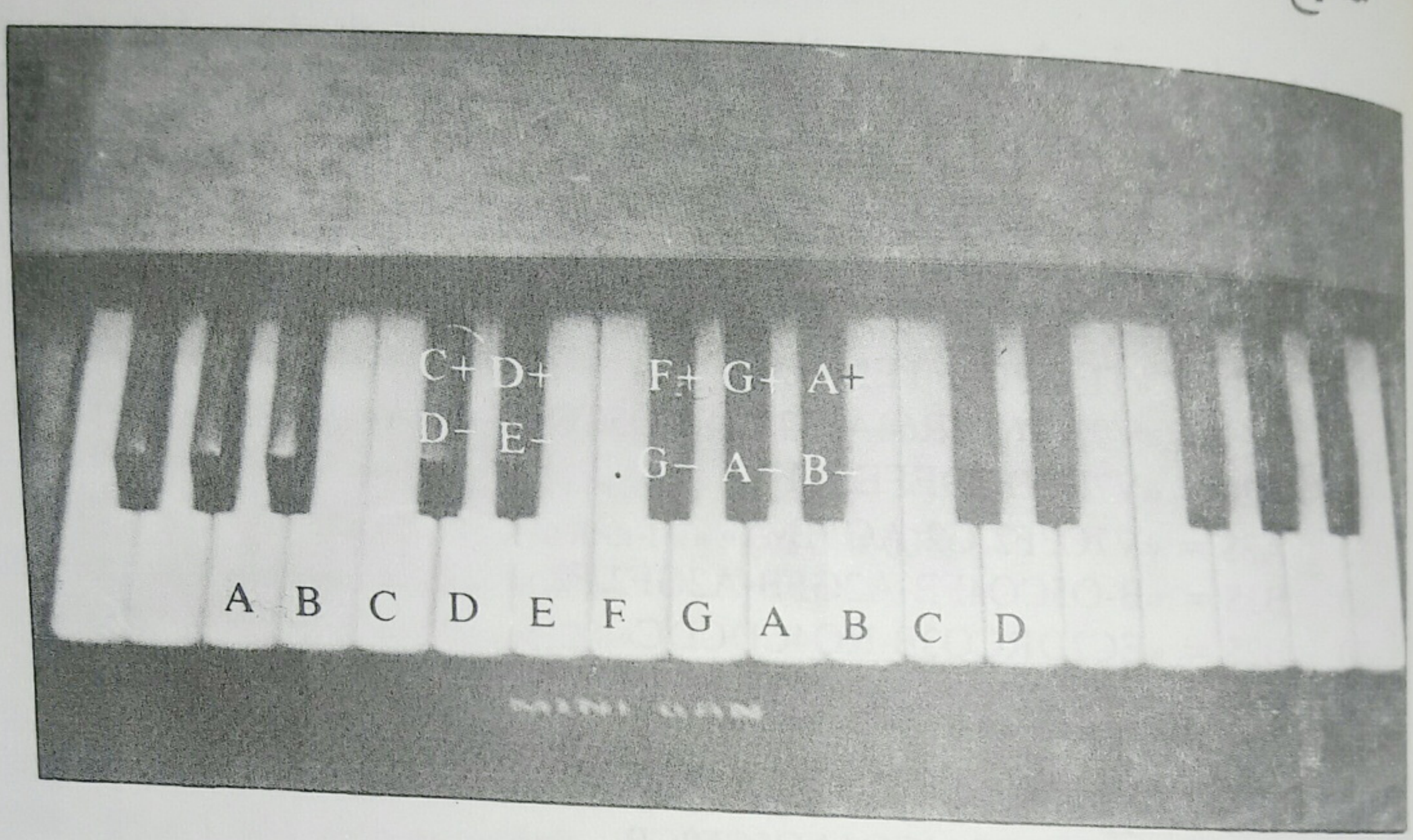
PLAY

اعزف

إذا كنت على دراية كافية بالموسيقى وبالنوتات الخاصة بها فلن تلاقي أية صعوبة في استخدام الرموز الخاصة بالعزف باستخدام الحاسب، ويستخدم البلاغ «PLAY» متبوعاً بعدد من الرموز الخاصة بالحاسب. الجدول أدناه يعطيك فكرة عن الرموز المستخدمة مع البلاغ وطرق استخدامها.

الرمز	وظيفة الرمز	الرمز	وظيفة الرمز
A	لا	V	صوت
B	سي	L	طول
C	دو	O	مدرج
D	ري	T	سرعة الايقاع
E	مي	R	استراحة
F	فا	N	عدد
G	صول	+, #	دييز
M	منغم	–	بيمول
S	شكل		

يعطيك الشكل التالي فكرة عن توزيع النوتات الخاصة بالحاسب على مفاتيح آلة البيانو.



نفذ البرنامج (C3) التالي ولاحظ كيفية استخدام البلاغ «PLAY» فيه. (قيمة «T» (TEMPO سرعة الايقاع) يتراوح بين 255,32).

```
0     REM *** C3 ***
10    REM *** DRINK TO ME ONLY MELODY
20    INPUT «TEMPO»; TEMPO
30    A$ = «V8AR64AR64AB-2R64B-O5C04B-AGA»
40    B$ = «B-O5CO4FB-A2GFB-A2GF2.F2.»
50    GOSUB 1000
60    GOSUB 1000
70    PLAY «V9O5 CR64CO4AO5CF2CR64
      CO4AO5CR64C2»
80    PLAY «CD2CR64CO4B-AR64A2.G2»
90    GOSUB 1000
100   END
990   REM SUBROUTINE
1000  PLAY «T = TEMPO; XA$;»
1010  PLAY «XB$;»
1020  RETURN
```

انتبه جيداً عند طباعة البرنامج فلا تخطىء بطباعة «0» بدلاً من «O» المستخدمة فيه. كذلك انتبه إلى مواقع الفوارز المنقوطة « ؛ » والنقاط « . ».

اطبع ونفذ البرنامج (C4) وستستمع إلى موسيقى شجية. حاول دراسة خطواته بشيء من التفصيل، لتستوعب استخدام كل رمز فيه.

```
5     TEM *** C4 ***
10    REM *** DRINK TO ME ONLY: MELODY
20    INPUT «TEMPO»; TEMPO
30    A1$ = «V9AR64AR64AB-2R64B-O5CO4B-AGA»
40    A2$ = «V7FEDCDEFEFEF»
50    A3$ = «V7O3F2.G2.AGFB-A»
60    B1$ = «B-O5CO4FB-A2GFB-A2GF2.F2.»
70    B2$ = «EC2DRCO3B-RO4CDCDC»
80    B3$ = «GAO2BO3C2.F2.C2.»
90    GOSUB 1000
100   GOSUB 1000
110   PLAY «V10O5CR64CO4AO5CF2CR
      64COAO5CR64C2», «V8EF2GF2GF2EGF»,
      «V8B-F2ED2EF2GA2»
120   PLAY «CD2CR64CO4B-AR64A2.G2.»,
      «E-DEFR64F2R64FR64F2.E2.», «GB-2AGO2GAB2.O3C2.»
130   GOSUB 1000
140   END
990   REM SUBROUTINE
1000  PLAY «T = TEMPO; XA1$,», «T = TEMPO; XA2$;»
1010  PLAY «XB1$;», «XB2$;», «XB3$;»
1020  RETURN
```

```
5     REM *** C5 ***
10    REM *** MUSIC PROGRAM ***
60    PLAY «t120», «t120», «t120»
70    PLAY «v10o514», «v8o414», «v8o214»
80    PLAY «e-o4b-g», «ge-o3b-», «e-12B-»
90    PLAY «f  # o512d», «o4do518co4L8b-aa-», «O1L4B-O4L8A-GG-F»
100   PLAY «o514», «o414», «o214»
110   PLAY «e-o4b-g», «ge-o3b-», «e-12b-»
120   PLAY «f # o512d», «o4do518co4L8b-aa-», «o114b-o4L8a-gg-f»
130   PLAY «14co4b-o5d», «14g12», «o214e-b-o3g»
```

```
140  PLAY «co4b-g», «14e-dd-», «o2e-b-o3e-»
150  PLAY «I1e-.», «o3b-ga-», «o212a.»
160  PLAY «g.», «o4cc # d», «b-.»
170  PLAY «I4b-o5cd»», «e-12e-», «14go312b-»
180  PLAY «fe-c», «14b-b-a», «14o2go3go2g-»
190  PLAY «o40a-.o518co4a-g», «R4cd-», «f14o3d-e»
200  PLAY «12f.», «r4cc», «o2fo3de-»
210  PLAY «14b-.o518do4b-a», «r4de-», «o212go314f#»
220  PLAY «12g.», «gd-c», «o214go3fe»
230  PLAY «t115», «t115», «t115»
240  PLAY «18o5cd12f», «a-o4dc», «o2a-o312f»
250  PLAY «t255», «t255», «t255»
260  PLAY «t115», «t115», «t115»
270  PLAY «o418fa-o512c», «c4e-a-f#», «o214b-o3fa-»
280  PLAY «t255», «t255», «t255»
290  PLAY «t120», «t120», «t120»
300  PLAY «14», «14», «14»
310  PLAY «e-o4b-g», «ge-o3b-», «e-12b-»
320  PLAY «f#o512d», «o4do518co418b-aa-», «o114b-o418a-gg-f»
330  PLAY «14e-o4b-g», «14gfe-», «o214e-o3gd-»
340  PLAY «fo512c», «e-18b-a-gf», «o2a-o312e-»
350  PLAY «14co4ge», «14edo3b-», «o214co3ec»
360  PLAY «cg-o5c», «o3go4d-e», «o2b-o3gc
370  PLAY «o412a.», «o3ao4e-d», «o212f.»
380  PLAY «a-.», «a-e-d», «o112v10b-.»
390  PLAY «18ga-14b-b», «18e-d14d-d», «o214v8e-18b-o3e-fg»
400  PLAY «o512c.», «cgf», «12a-.»
410  PLAY «18de-14ff», «fgg», «14g18o3dgab»
420  PLAY «12g.», «o5ed-c», «o414cfe»
430  PLAY «14e-co4a-», «o4a-a-e», «o312f.»
440  PLAY «fo5co4d», «ce-o3a-», o214b-o3fo2f-»
450  PLAY «v1111e-», «v7o418e-fgb-o514ce-», «e-b-b-e-»
460  END
```

البيسك العربي

نظام البيسك العربي هو مفهوم جديد للحاسبات الثنائية اللغة (العربي/ الانكليزي)، والنظام من تطوير الشركة العالمية بالاشتراك مع مايكروسوفت. ويتم تحويل الحاسب إلى هذا النظام باستخدام الكارتردج (الكاسيت) الخاص بالبيسك العربي.

ويتيح لنا النظام استخدام الأوامر والبلاغات والدوال الخاصة بلغة البيسك وباللغة العربية. وقد استخدمنا في فصول الكتاب السابقة الإيعازات العربية المرادفة لإيعازات (MSX BASIC). وطريقة استخدامها مشابهة تماماً لاستخدام إيعازات (MSX BASIC). لاحظ البرنامج التالي:

```
١٠ س = ٥ : ص = ١٠
٢٠ ح = س + ص * ٢٠
٣٠ اطبع ص
```

يتغير عمل مفاتيح الدوال عند استخدام الكارتردج الخاص بالبيسك العربي ويمكن ملاحظة ذلك في أسفل الشاشة.

يمكن لنا كذلك ترجمة البرامج الخاصة بالبيسك العربي إلى برامج خاصة بـ (MSX BASIC)، وذلك باستدعاء الأمر «ترجم»، ويتم ذلك بالضغط على مفتاح الدوال (F7).

ولغرض تسهيل عملية طباعة البرامج يمكن استخدام المفتاح «GRAPH» مع المفاتيح الأخرى للحصول على بلاغات ودوال صخر العربية، دون الحاجة الى طباعتها. الجدول أدناه يوضح عمل كل مفتاح مع مفتاح الـ «GRAPH»:

المفتاح	GRAPH	GRAPH + SHIFT	المفتاح	GRAPH	GRAPH + SHIFT
أ	اذا	إذن	ن	نفذ	نظف
ب	بيان	باق	ه	هاش	هاي
ت	تالي	ترقيم	و	والا	
ث	ثاحمل		لا	لا نقطة (	لا تتبع
ج	جديد	جدول (	ي	يمين $ (	
ح	حمل "	حفظ "	١	⊥	
خ	خال (٠)	خطوة	٢	⊤	
د	دخل	دائرة	٣	⊥	
ذ	رمز (	رجع	٤	⊤	
ر	سرد	سطر	٥	+	
س	شاشة	شمال $ (	٦	⊣	
ش	صبغ (	صح (	٧	—	
ص	ضع	_	٨	⌐	
ض	طاطبع	طاسرد	٩	¬	
ط	عزف		٠	└	
ع	غمد (		_	┘	
ع	فتح	فراغ $ (	=	×	
غ	قصد	قرأ	\	π	
ف	لون	لونقطة (			
ق	مفتاح	ملفات			
ل					
م					

يمكن لك ترجمة كافة البرامج الموجودة في الكتاب، وكذلك طباعتها باستخدام (MSX BASIC) مع استخدام كارتردج البيسك العربي، ثم استخدام الأمر «ترجم» لترجمتها.

يترجم الأمر «ترجم» كافة الأوامر والبلاغات المستخدمة في لغة البيسك، ولكن يبقى عمله قاصراً للمعلومات والرموز التي ترد بين حاصرتين علويتين،

كالتي ترد في جمل الملحوظات أو جمل الإدخال، والتي ترد مع البلاغين «PLAY» و«DRAW»، لذلك يتوجب علينا إعادة ترجمتها. اطبع البرنامج التالي ولاحظ ذلك:.

```
10    X$ = «العالمية»
20    Y$ = «AL ALAMIA»
30    PRINT X$, Y$
```

اضغط الآن على المفتاح (F7) ثم اسرد البرنامج ستلاحظ البرنامج التالي:

```
١٠  ظ $ = «IjedGYdG»
٢٠  ع $ = «ء ج ء ج ء حة ء ب»
٣٠  اطبع ظ $، ع $
```

لاحظ أن الرموز التي وردت في السطرين 10 و 20 غير مفهومة لذلك يتوجب إعادة كتابتها.

اطبع الآن البرنامج التالي:

```
١٠  شاشة ٢
١٥  نقطة (١٠٠، ١٠٠)
٢٠  ارسم «ي ٢٠ت ٢٠ش٢٠ق٢٠»
٣٠  اقصد ٣٠
```

نفذ البرنامج سيظهر لك على الشاشة رسم مربع، اضغط الآن على المفتاحين (STOP + CTRL) سوية لإيقاف البرنامج ثم اضغط على المفتاح (F7) لترجمته، اسرد البرنامج لتلاحظ التالي:

```
10    SCREEN 2
15    PESET (100, 100)
20    DRAW «j02j02T02a02»
30    GOTO 30
```

نفذ البرنامج لتلاحظ العبارة التالية:

```
Illegal function call in 20
```

لذلك يجب استبدال الرموز في السطر 20 إلى الرموز المناسبة، ليصبح البرنامج كالآتي:

```
10  SCREEN 2
15  PSET (100, 100)
20  DRAW «R20D20C20U20»
30  GOTO 30
```

اضغط الآن على المفتاح (F7) مرة أخرى واسرد البرنامج لتلاحظ النتيجة التالية:

```
١٠  شاشة ٢
١٥  نقطة (١٠٠، ١٠٠)
٢٠  ارسم «ر٢٠ؤ٢٠ج٢٠ص٢٠»
٣٠  اقصد ٣٠
```

وهذا أيضاً يعطي إشارة بالخطأ في السطر 20 إذا ما نفذناه، لذلك يجب إعادة كتابة الرموز بالصورة الصحيحة.

الجدول أدناه يوضح الرموز الخاصة بصخر بيسك العربي و (MSX BASIC) والمستخدمة مع البلاغ «DRAW».

وظيفة الرمز	صخر بيسك (مباشر)	صخر بيسك (ترجم)	MSX-BASIC
فوق	ف	ص	U
تحت	ت	ؤ	D
شمال	ش	ج	L
يمين	ي	ز	R
حركة	ح	ح	M
شمال شرق	ق	إ	E
جنوب شرق	ك	ىء	F
جنوب غرب	ه، ة		G
جنوب شرق	ى	ب	H
زاوية	د	ء	A
حركة بدون رسم	و	آ	B

[الجدول السابق ـ تتمة]

وظيفة الرمز	صخر بيسك (مباشر)	صخر بيسك (مترجم)	MSX-BASIC
لا تغير الإحداثيات	لا	خ	N
نفذ	ن	ظ	X
لون	ل	أ	C
مقياس	م	س	S

كذلك الأمر مع البلاغ «PLAY» حيث يجب إبدال الرموز عند إجراء عملية الترجمة. ويمكن إجراء عملية الإبدال هذه بالرجوع إلى الجدول التالي:

وظيفة الرمز	صخر بيسك (مباشر)	صخر بيسك (مترجم)	MSX-BASIC
لا	ل	ء	A
سي	س	آ	B
دو	د	أ	C
ري	ر	ؤ	D
مي	م	إ	E
فا	ف	ىء	F
صول	ص	ا	G
منغم	غ	ح	M
شكل	ش	س	S
صوت	ت	ض	V
طول	ط	ج	L
مدرج	ج	د	O
سرعة الإيقاع	و	ش	T
استراحة	ح	ر	R
عدد	ن	خ	N
دييز	# و +	# و +	# و +
بيمول	–	–	–
نقطة			

يجب ملاحظة التالي:

(أ) عند كتابة النوتة بـ (MSX-BASIC) بغرض ترجمتها فيجب كتابة الحروف من (G-A) بالحروف العالية (CAPITAL LETTERS) وكتابة الحروف الباقية بالأحرف العادية (SMALL LETTERS) لكي يمكن تنفيذها بصخر بيسك.

(ب) عند ترجمة البلاغات فإن ترتيب الأرقام يكون مقلوباً مثل (١٢٣) ستصبح (٣٢١) لذا يجب تعديل وضع الأرقام عند تنفيذ ترجم وقبل تنفيذ البرنامج (راجع الفقرة الأولى من هذا الباب).

(ج) عند تنقيط نوتة يجب وضع النقطة (الفاصلة الحسابية) قبل الرقم (مثال: أعزف «در٨»).

أدناه بعض البرامج العربية التي يمكن تنفيذها، لتكون مراجعة للفصول السابقة.

```
5' T1                                          ٥' ع١
10' DRAW EMPTY RECTANGLE                       ١٠' رسم مستطيل فارغ
20' ROTATE ON ITSELF                           ٢٠' دورانه حول نفسه
120 LET D = 1                                  ١٢٠ دع ؤ = ١
130 COLOR 15, 1, 1                             ١٣٠ لون ١٥، ١، ١
140 SCREEN 2                                   ١٤٠ شاشة ٢
170 LET D = D + 1: IF D > 15 THEN 120          ١٧٠ دع ؤ = ؤ + ١: إذا ؤ > ١٥ إذن ١٢٠
180 LET S = 0                                  ١٨٠ دع س = ٠
210 LINE (S, 0) – (0, 255 – S), D              ٢١٠ سطر (س، ٠) - (٠، ٢٥٥ - س)، ؤ
220 LINE – (255 – S, 255), D                   ٢٢٠ سطر - (٢٥٥ - س، ٢٥٥)، ؤ
230 LINE – (255, S), D                         ٢٣٠ سطر - (٢٥٥، س)، ؤ
240 LINE – (S, 0), D                           ٢٤٠ سطر - (س، ٠)، ؤ
270 S = S + 10                                 ٢٧٠ س = س + ١٠
280 IF S < 255 THEN 210                        ٢٨٠ إذا س < ٢٥٥ إذن ٢١٠
290 GOTO 170                                   ٢٩٠ اقصد ١٧٠
310 END                                        ٣١٠ نهاية
```

```
5    'T2
120 COLOR 15, 1, 1
130 SCREEN 2
160 FOR E = 1 TO 15
190 FOR D = 2 TO 15
220 LINE (E + D, E + D) - (E
    * D, E * D), D, BF
230 NEXT D
240 NEXT E
260 GOTO 260
280 END
```

```
٥  ' ع٢
١٠ ' رسم مربعات ملونة
١٢٠ لون ١٥، ١، ١
١٣٠ شاشة ٢
١٦٠ من م = ١ إلى ١٥
١٩٠ من ل = ٢ إلى ١٥
٢٢٠ سطر (م + ل، م + ل) - (م *
    ل، م * ل)، ل، صم
٢٣٠ تالي ل
٢٤٠ تالي م
٢٦٠ اقصد ٢٦٠
٢٨٠ نهاية
```

```
5    ' T3
30   SCREEN 2
40   Y = 40
70   S = 50: U = 210
80   GOSUB 180
110  U = 130: S = 130
120  GOSUB 180
150  GOTO 150
180  FOR E = 0 TO 40 STEP 3
190  LINE (S + E, E + 10) - (S +
     E, E + Y + 10)
200  LINE (U - E, E + 10) - (U
     - E, E + Y + 10)
210  LINE (S + E, 90 + 2 * Y -
     E) - (S + E, 90 + Y - E), 1
220  LINE (U - E, 90 + 2 * Y -
     E) - (Y - E, 90 + Y - E), 1
230  NEXT
240  RETURN
```

```
٥  ' ع ٣
٣٠  شاشة ٢
٤٠  ع = ٤٠
٧٠  س = ٥٠ : ص = ٢١٠
٨٠  تفرع ١٨٠
١١٠ ص = ١٣٠ : س = ١٣٠
١٢٠ تفرع ١٨٠
١٥٠ اقصد ١٥٠
١٨٠ من م = ٠ إلى ٤٠ خطوة ٣
١٩٠ سطر (س + م، م + ١٠) - (س +
    م، م + ع + ١٠)
٢٠٠ سطر (ص - م، م + ١٠) - (ص
    - م، م + ع + ١٠)
٢١٠ سطر (س + م، ٩٠ + ٢ * ع -
    م) - (س + م، ٩٠ + ع - م)، ١
٢٢٠ سطر (ص - م، ٩٠ + ٢ * ع -
    م) - (ع - م، ٩٠ + ع - م)، ١
٢٣٠ تالي م
٢٤٠ ارجع
```

ملحق (2) إيعازات البيسك

يشمل الملحق جميع الإيعازات المستخدمة في لغة البيسك والخاصة بـ (MSX) موضحاً الإيعاز، الغرض منه، وصيغة كتابته. إضافةً إلى ذلك سنجد أيضاً إشارة للبرنامج الذي استخدم فيه الإيعاز ضمن الكتاب إن وجد، وكذلك الصفحة.

تستخدم جميع الإيعازات في الحاسب «صخر» كما هي، وكذلك في الحاسب «NEC»، فيما عدا تلك المعلمة بعلامة «*»، فإنها لا تستخدم مع الحاسب «NEC» أو تستخدم بصيغة أخرى، وهذا ما سنلاحظه في الملحق (3) الخاص بالحاسب «NEC». ما موجود بين [ ] يكون وضعه اختيارياً.

مطلق ABS

.....

لإعطاء القيمة المطلقة لبيان عددي.

```
ABS 10 = 10
ABS −1000 = 1000
```

رمز ASC

.....

لإعطاء قيمة الرمز (ASCII) المناظر للحرف الأول للمقطع.

```
10    A$ = «FRIDAY»
20    B = ASC (A$)
30    PRINT B
RUN
70
O.K
```

قظا		ATN

لإعطاء مقلوب ظل الزاوية.

```
PRINT ATN(1.5)
.98279372324731
```

ذاتي *	16	AUTO

لإعطاء أرقام السطور للبرنامج ذاتياً

AUTO [رقم سطر]، [عدد ثابت]

يمكن استخدامه مباشرة ليبدأ ليرقم السطر الأول بالرقم 10 وبزيادة 10 أيضاً لأرقام الأسطر الأخرى. كذلك يمكن تحديد رقم السطر الأول، ومقدار الزيادة.

بيب *	BEEP

لإصدار صوت شبيه بصوت (بيب)

ثنا $	BIN$

لتحويل بيان عددي إلى الثنائي.

```
10    PRINT BIN$(255)
20    PRINT BIN$(9999)
RUN
11111111
10011100001111
```

.. نادي	23	CALL

لتنفيذ أمر موجود في الـ «ROM» الإضافي.

لضعف CDBL

.....

لتحويل بيان عددي إلى بيان ذي دقة مضاعفة.

حرف $ B4 112 CHR$

.....

لإعطاء الحرف المناظر لرقم معين (راجع جداول الرموز)

```
PRINT CHR$(65) <CR>
      A
```

لصح CINT

.....

لتحويل بيان عددي إلى بيان عددي صحيح.

```
PRINT CINT (123.432)    <CR>
      123
```

دائرة A51 93 CIRCLE

.....

لرسم دائرة، شكل بيضوي أو قوس من دائرة.

CIRCLE (س، ص)، ق، ل، ب، ن، آ

حيث تمثل (س،ص): إحداثيات المركز

ق: نصف القطر

ل: لون الرسم

ب،ن: زاويتي البداية والنهاية (نصف قطرية)

آ: النسبة بين قطري الشكل البيضوي.

نظف CLEAR

.....

لمسح حيز محدود من الذاكرة

```
CLEAR A
```

كاحمل 16, 21 CLOAD

لتحميل برنامج على شريط الكاسيت.

CLOAD «اسم الملف»

كاحمل؟ 21 CLOAD?

لمقارنة البرنامج المخزون على شريط الكاسيت مع الموجود في الذاكرة.

CLOAD? «اسم الملف»

مقفل CLOSE

قفل ملف سبق فتحه باستخدام البلاغ «افتح»

CLOSE # رقم الملف

امسح 42 A1 CLS

يستخدم لمسح ما موجود على الشاشة

CLS

لون * 84 A41 COLOR

يستخدم لتحديد الألوان على الشاشة.

لون الإطار، لون الخلفية، لون الكتابة COLOR

استمر 16 CONT

لإستئناف تنفيذ البرنامج بعد توقفه.

CONT

COS

جتا

. . . . . .

لإعطاء جتا الزاوية

COS (A)

```
PRINT COS (1.4445432)      <CR>
.1259179846524
```

CSAVE

كاحفظ

. . . . . .

يستخدم لحفظ البرامج على شريط الكاسيت

CSAVE «اسم الملف»

CSNG

لفرد

. . . . . .

لتحويل بيان عددي ذي دقة عالية إلى بيان ذي دقة اعتيادية.

```
PRINT COS (0.7656)      <CR>
.72096670541357
PRINT CSNG (COS(0.7656) )      <CR>
.720967
```

CSRLIN

موسطر

. . . . . .

يعطي رقم السطر الذي تقع عليه المشيرة

CSRLIN

DATA A21 61

بيان

. . . . . .

لإمداد قائمة بيانات داخل البرنامج لقرائتها بواسطة البلاغ «READ»

عدد من البيانات DATA

تفصل البيانات فيما بينها بفارزة « ، ».

كضعف DEFDBL

لتحديد دقة متغير بالدقة المضاعفة

```
DEFDBL X
```

عرف دالة DEF FN

يستخدم لتعريف دالة قبل استخدامها

```
DEF FN (Y) = X ^ 3
```

كصح DEFINT

لتحديد نوع المتغير من كونه عدد صحيح أم لا.

كفرد DEFSNG

لتحديد دقة متغير بالدقة المفردة وذلك بدلالة أول حرف.

الغ DELETE

لإلغاء سطر معين من البرنامج أو عدد من السطور

DELETE (رقم سطر) - (رقم سطر)

بعد DIM A20 58

لتحديد أبعاد البيانات في متغير مصفوفي

```
DIM A(5,5,5)
```

DRAW A58 103 ارسم

لرسم أشكال باستخدام رموز معينة

«مـقطع من الرموز» DRAW

END 40 نهاية

ينهي تنفيذ البرنامج

END

EOF نهام

للتأكد من وصول عملية قراءة ملف إلى نهايته

EOF اسم ملف

EOF(4) = − 1 THEN «END OF FILE»

ERASE امح

لمسح متغير مصفوفي واحد، أو أكثر

ERASE A

A يمثل متغير مصفوفي.

ERL خطا س

تعطي رقم السطر الذي حدث عنده الخطأ.

ERR خطا ر

تعطي رقم آخر خطأ حدث بالبرنامج.

خطا ERROR

.....

يدخل البرنامج بالقصد في حالة خطأ من نوع معين وذلك لمحاكاة حالة حدوثه ومعرفة كيفية تصرف البرنامج تجاهه.

رقم الخطأ ERROR

اس EXP

.....

يعطي قيمة الأس الطبيعي e ^ س.

EXP A

بتر FIX

.....

لإعطاء الجزء الصحيح لعدد معطى

FIX (1.87) = 1

FIX (−12.88) = −12

من.. إلى.. خطوة.. تالي FOR.. TO.. STEP.. NEXT

.....

لتكرار تنفيذ فقرة معينة من البرنامج.

خال FRE

.....

لبيان عدد الوحدات الخالية من الذاكرة

يبين المقدار الباقي من الذاكرة للبرنامج. PRINT FRE(0)

يبين المقدار الباقي من الذاكرة للمقطع. PRINT FRE (« »)

GOSUB.... RETURN

تفرع.... ارجع

يستخدم لنقل البرنامج إلى برنامج فرعي (روتين فرعي)

GOSUB (رقم سطر)

RETURN

GOTO

اقصد

لتحويل مسار البرنامج إلى سطر معين.

GOTO رقم سطر

HEX$

ستع $

لتحويل بيان عددي إلى الستعشري

```
PRINT HEX$(255)     <CR>
FF
O.K
```

IF... THEN... ELSE +A32 71 69 إذا... إذن... وإلا

لتفرع البرنامج إلى سطر معين عند حدوث شرط معين.

```
10    IF A > B THEN 20 ELSE GOTO 30
20    PRINT «A > B»
30    PRINT «A < = B»
```

INKEY$ A15 54 كشف $

السماح لإدخال حرف أو رمز واحد من لوحة المفاتيح

INPUT A11 52 ادخل

يستخدم لإدخال بيانات من لوحة المفاتيح

INPUT [تنويه] المتغير؛

INPUT # ادخل #

يقرأ بيانات من ملف مفتوح، ثم تعينه لمتغير معين

INPUT # متغير، رقم الملف

INPUT $ A17 57 ادخل $

يستقبل عدد معين من الحروف من لوحة المفاتيح

INPUT $ (N)

حيث تمثل (N) عدد حروف المقطع المطلوب.

INSTR فيحزم

تعطي موقع مقطع جزئي معطى داخل مقطع آخر.

INT صح

تعطي أكبر عدد صحيح أصغر من البيان المعطى

INT (10.111111) = 10
INT (−1.234) = −2

INTERVAL ON/OFF/STOP برهة نعم / كلا / قف

لبدء التعامل مع أو إبطال مفعول أو الإبقاء على حدث الاعتراض الذي يحدثه الموقت الداخلي للحاسب. مثلاً:

10 ON INTERVAL = 60 GOSUB 1000

KEY

مفتاح

يعين مقطع من الحروف لأحد مفاتيح الدوال

KEY 1, «PRINT»

الضغط على المفتاح «1» يؤدي الآن إلى طباعة العبارة «PRINT».

ويمكن أن يكون ذلك أي مقطع نحتاجه بكثرة في البرنامج، ليسهل علينا عملية طباعته. يمكن كذلك إضافة CHR$(13) إلى المقطع، ليغنينا عن الضغط على مفتاح الإدخال.

KEY 2, «LIST» + CHR$(13)

الضغط على المفتاح «2» يؤدي إلى سرد البرنامج مباشرة، كما لو كنا قد ضغطنا على المفتاح «4» ثم على مفتاح الإدخال <CR>.

KEY LIST

مفتاح أسرد

لسرد محتويات مفاتيح الدوال العشرة

KEY LIST

KEY ON/OFF

مفتاح نعم / كلا

لإظهار أو إخفاء أسماء مفاتيح الدوال من أسفل الشاشة.

KEY(N) ON/OFF/STOP

مفتاح (ن) نعم / كلا / قف

لبدء التعامل مع حدث الاعتراض الناجم عن ضغط أحد مفاتيح الدوال.

LEFT $

شمال $

تقوم باستخلاص عدد من الحروف مأخوذة من الجانب الأيسر للمقطع.

LEFT $ (المقطع، A)

A تمثل عدد ثابت من 0 إلى 255، متغير عددي، أو صيغة عددية.

LEN طول

.....

تبين عدد حروف المقطع المعطى

LEN (A)

A تمثل متغير مقطعي

```
10    A$ = RIYADH
20    B = LEN(A$)
30    PRINT B
RUN
6
O.K
```

LET

دع

.....

لإعطاء قيمة لمتغير عددي أو مقطعي.

```
LET A = 100
LET A$ = «WEEK»
```

ويمكن الاستغناء عنها في نظام (MSX).

LINE A46 88

سطر

.....

يرسم خط بين نقطتين أو مربع فارغ أو مملوء

LINE (X1, Y1) – (X2, Y2), C, BF

X1, Y1 إحداثيات النقطة الأولى

X2, Y2 إحداثيات النقطة الثانية

C لون الرسم

B لرسم مربع فارغ يمثل الخط الواصل بين النقطتين القطر المائل

BF لرسم مربع مملوء.

LINE INPUT

سطر ادخل

لإدخال أي مقطع من الحروف من لوحة المفاتيح

```
10   LINE INPUT «WHAT SHALL I DO NOW?; S$
20   PRINT «O.K YOU»; S$
RUN
WHAT SHALL I DO NOW? GET LAMP, SWORD, AND WATER
O.K YOU GET LAMP, SWORD AND WATER
```

LINE INPUT #

سطر ادخل #

يقرأ مقطع بيانات من ملف ويعينه إلى متغير مقطعي

```
LINE INPUT # B, A$
```

B تمثل رقم الملف المفتوح، و A$ تمثل المتغير المقطعي.

LIST 16, 40

اسرد

لسرد نص البرنامج الموجود في ذاكرة الحاسب على الشاشة.

LIST

لسرد البرنامج الموجود في الذاكرة

LIST (رقم سطر معين)

لسرد سطر معين من البرنامج

لسرد جزء من البرنامج يبدأ بالسطر ذي A إلى السطر ذي الرقم B

```
LIST A-B
```

LLIST 41

طاسرد

لسرد نص البرنامج على الطابعة.

لا تختلف صيغه عن صيغ الإيعاز «LIST».

حمل LOAD

لتحميل برنامج تم حفظه برموز ASCII

LOAD «اسم الملف: مميز الجهاز»، R

إذا ما وضعنا الرمز «R» في النهاية يؤدي ذلك إلى تنفيذ البرنامج حال تحميله دون الحاجة إلى إعطاء الأمر «RUN».

حدد LOCATE A4 47

يحرك المشيرة إلى وضع معين على الشاشة

LOCATE X, Y

لو LOG

تعطي قيمة اللوغرتم الطبيعي

LOG (A)

A تمثل عدد ثابت، متغير عددي، أو تعبير عددي.

طاموضع LPOS

تعطي الموضع الذي يكون عنده رأس الطابعة

PRINT LPOS (X)

طاطبع LPRINT 41

لطباعة مقطع معين على الطابعة

LPRINT «ALI»

LRPINT X

LPRINT USING

طاطبع مستخدماً *

.....

لطبع بيانات على الطابعة بشكل معين.

راجع PRINT USING

MAX FILES

أكبر ملفات *

.....

لتحديد عدد الملفات (أكبر عدد)

MAXFILES = A
A = 0 – 15

MERGE

ادمج

.....

لدمج ملف محفوظ برمز (ASCII) مع برنامج في ذاكرة الحاسب.

MERGE «اسم البرنامج: مميز الجهاز»

MID $

وسط $

.....

تستخلص مقطع جزئي من مقطع معين.

MOD

باقي *

.....

لإعطاء باقي القسمة.

20 MOD 3 = 2

MOTOR

محرك * 20

.....

لتشغيل وإيقاف محرك جهاز التسجيل.

MOTOR ON
MOTOR OFF

جديد

NEW

يمسح البرنامج الموجود في ذاكرة الحاسب تمهيداً لإدخال برنامج جديد.

NEW

ثماني *

OCT$

لتحويل بيان عددي إلى القيمة الثمانية.

```
10    PRINT OCT$ (8)
20    PRINT OCT$ (8 ^ 4)
30    PRINT OCT$ (8 ^ 4 - 1)

run
10
10000
7777
```

نعم.... تفرع

ON.... GOSUB A27 67

لتوجيه البرنامج لتنفيذ برنامج فرعي وذلك حسب قيمة التعبير الوارد في البلاغ

ON X GOSUB 100, 200, 500

إذا X = 1 تفرع إلى السطر 100
إذا X = 2 تفرع إلى السطر 200
إذا X = 3 تفرع إلى السطر 500

نعم.... اقصد

ON.... GOTO A26 66

يقود البرنامج إلى سطر معين وحسب قيمة التعبير العددي.
راجع البلاغ ON... GOSUB

ON ERROR GOTO * نعم خطأ اقصد

.....

يقود البرنامج إلى سطر معين عند حدوث خطأ ما.

ON ERROR GOTO رقم سطر

ON INTERVAL * نعم برهة تفرع

.....

لبيان برنامج فرعي تم التفرع إليه عند حدوث اعتراض من الموقت الداخلي للحاسب.

ON INTERVAL = 500 GOSUB 40

ON KEY GOSUB * نعم مفتاح تفرع

.....

للانتقال إلى برنامج فرعي معين عند الضغط على أحد مفاتيح الدوال.

ON SPRITE GOSUB B13 135 * نعم شبح تفرع

.....

للانتقال إلى برنامج فرعي معين عند تصادم شبحين.

ON SPRITE GOSUB رقم سطر

ON STOP GOSUB * نعم قف تفرع

.....

للانتقال إلى برنامج فرعي معين عند الضغط على المفتاحين «CTRL + STOP»

ON STOP GOSUB رقم سطر

ON STRIG GOSUB * نعم زناد تفرع

.....

للانتقال إلى برنامج فرعي معين عند الضغط على قضيب المسافة أو زناد عصا التحكم.

ON STRIG GOSUB رقم السطر

افتح OPEN

لفتح ملف وتحديد طور استخدامه.

OPEN «اسم الملف مميز الجهاز» FOR طور # الملف رقم

الطور INPUT للكتابة عليه.

OUTPUT للقراءة منه.

خارج OUT

لإخراج بيان بطول بايت واحد إلى أحد منافذ الإدخال/ الإخراج. مثلاً:

OUT 1, 166

لوح PAD

لإعطاء الحالة التي عليها لوح الرسم

PAD(n)

n = 1 إلى 3 إذا ما ربط اللوح إلى وصلة العصى 1

1- إذا لمس اللوح و 0 إذا لم يلمس اللوح	PAD(0)
لتحديد الموقع على المحور «X» (0-255)	PAD(1)
لتحديد الموقع على المحور «Y» (0-255)	PAD(2)
1- إذا ما ضغط على مفتاح اللوح	PAD(3)

اصبغ A56 100 PAINT

لصبغ منطقة مغلقة بلون معين

PAINT (X,Y), C1, C2

X, Y تمثل إحداثيات نقطة داخل الشكل

C2, C1 تمثلان لون الصبغة والإطار الخارجي لها.

PDL

بدال

. . . . . .

لإعطاء الحالة التي عليها بدال اللعب، ويمكن ربط (12) بدال على الحاسب.

PDL (n)

n عدد صحيح من 1 إلى 12.

PEEK

غمد

. . . . . .

ترجع مضمون الذاكرة عند عنوان معين.

PEEK (عدد صحيح)

PLAY

اعزف

. . . . . .

يستخدم لعزف الموسيقى.

PLAY «مقطع من الرموز»

اعزف (N)

. . . . . .

يختبر إذا ما كانت الموسيقى تصدر من القناة «N» أم لا.

N = 1, 2, 3

POINT

لونقطة

. . . . . .

تستخدم لتحديد نقطة معينة.

POINT (X,Y)

دمغ

POKE

هو الوجه المقابل لغمد، حيث يكتب بياناً معيناً في الذاكرة.

عدد، السجال POKE

موضع ( )

POS ( )

تعطي الحداثي السيني للموضع الذي تشغله المشيرة.

لانقطة *

84

PRESET

يرجع لون نقطة معينة إلى لون الخلفية (أي يمسحها).

PRESET (X,Y),C

X, Y تمثل إحداثي النقطة، إذا استخدمنا (C) فيصبح عملها مشابهاً لعمل «PSET».

اطبع

25

PRINT

لإظهار البيانات على الشاشة.

اطبع #

PRINT #

لكتابة البيانات على ملف تمَّ فتحه.

التعبير، رقم الملف # PRINT

PRINT USING اطبع مستخدماً *

لإظهار البيانات على الشاشة بشكل معين.

التعبير؛ رمز الصيغة PRINT USING

الرمز	التأثير
!	إظهار الحرف الأول من كل مقطع فقط
& ن &	إظهار عدد من الحروف مساوي لـ ن + 2 من كل مقطع
δ	لإظهار جميع حروف المقاطع المعطاة
-	لإضافة علامة السالب للعدد السلبي
.#	لتحديد خانات الأرقام الصحيحة والخانات العشرية
+	تعني ضرورة إضافة علامة «+» للأرقام الموجبة والـ «-» للسالبة
* *	لإحلال علامة «*» بدلاً من خانات الفراغ على يسار الرقم المعطى

PRINT # USING اطبع # مستخدماً *

لكتابة بيانات على ملف مفتوح وبصيغة معينة.

PSET A43 86 نقطة

لوضع نقطة في مكان محدد من الشاشة وبلون معين.

PSET (X, Y), C

X, Y	تمثل إحداثي النقطة
C	يمثل لونها

ضع شبح * B4 112 PUT SPRITE

لإظهار شكل شبحي معين في موضع معين.

PUT SPRITE A, (X, Y), C, B

A تمثل رقم المستوى 0-32

X, Y الإحداثي السيني والصادي، و C اللون.

B = 0 – 255 للشبح حجم 8 × 8، و 0 – 63 للشبح حجم 16 × 16

اقرأ A21 61 READ

لقراءة البيانات الموجودة في البرنامج ضمن البلاغ «DATA»

READ A

A يمثل متغير عددي أو حزمي.

ملحوظة 41 REM

لكتابة ملحوظة داخل البرنامج، وكل ما يكتب خلفها لا يؤثر على عمل البرنامج، ويمكن الاستعاضة عنها بـ «'».

ترقيم RENUM

لإعادة ترقيم سطور البرنامج، وإذا لم نحدد الأرقام المطلوبة للسطور سيكون رقم السطر الأول هو 10 وأرقام السطور الأخرى تكون بفارق 10 أيضاً.

RENUM الزيادة، السطر الثاني، السطر الأول

عاود A22 62 RESTORE

لإعادة قراءة البيانات الواردة في البلاغ «DATA»

واصل * RESUME

يعيد تنفيذ البرنامج الأصلي بعد تنفيذ أحد البرامج الفرعية الخاصة بالأخطاء.

يمين $ RIGHT $

لإستخلاص عدد من الحروف مأخوذة من الجانب الأيمن لبيان مقطعي.

RIGHT$ (A، المقطع)

A تمثل عدد ثابت بين الـ 0 و 255، متغير عددي، أو صيغة عددية.

عشو RND A35 74

تعطي رقم عشوائي بين الـ «0» والـ «1».

RND(1)

نفذ RUN

لتنفيذ البرنامج، ويمكن تنفيذ جزء من البرنامج وذلك بتحديد السطر الأول لهذا الجزء.

RUN [رقم سطر]

احفظ SAVE

لحفظ برنامج البيسك برمز ASCII

SAVE «اسم الملف: مميز الجهاز»

SCREEN A38 79 * شاشة

. . . . . .

SCREEN [A,B,C, D, E] لتحديد :

1 ـ طـــور الشـــاشـــة A: 0 طور الكتابة 40 × 24

1 طور الكتابة 24 × 32

2 طور الدقة العالية

٣ طور الألوان المتعددة

2 ـ حـــجـــم الشبـــح B: 0 شبح 8 × 8 غير مكبر

1 شبح 8 × 8 مكبر

2 شبح 16 × 16 غير مكبر

3 شبح 16 × 16 مكبر

3 ـ صـــوت المـــفتـاح C: 0 لاصوت

X صوت (X أي رقم غير الصفر)

4 ـ معدل نقل البيانات D: 1 1200 بورد

2 2400 بورد

5 ـ نـــوع الطـــابعـــة E: 0 طابعة MSX

1 طابعة صخر معربة

2 طابعة معربة تختار شكل الحرف تلقائياً

3 طبع صورة مطابقة لما موجود على الشاشة

إشارة

SGN

. . . . . .

. . . . . .

تعطي القيمة «1» للإشارة الموجبة و «−1» للسالبة و «0» للصفر.

جا

SIN

. . . . . .

. . . . . .

تعطي قيمة جيب الزاوية (نصف قطرية)

```
PRINT SIN (1)
0.84147098480792
```

SOUND صوت

لتوليد مؤثرات صوتية معينة.

تعبير، رقم السجالة SOUND

SPACE $ فراغ $ *

توليد مقطعاً بأي عدد من خانات الفراغ.

SPACE$ (A)

255 − 0 = A

SPC فرغ

لإخراج أي عدد من خانات الفراغ.

```
10    A$ = «ALI»: B$ = «RIYADH»
20    PRINT A$; SPC(10); B$
      run
ALI         RIYADH
```

SPRITE ON / OFF / STOP B13 135 * شبح نعم / كلا / قف

للتعامل مع، أو إبطال مفعول والإبقاء على حدث الاعتراض الناجم عن تصادم شبحين.

SPRITE ON
SPRITE OFF
SPRITE STOP

SPRITE $ B4 112 * شبح $

لتحديد الأشكال الشبحية.

SPRITE$ (N) = مقطع من الرموز

جذر SQR

لإعطاء الجذر التربيعي لعدد ما

SQR (N)

N عدد صحيح موجب.

عصا STICK

لإعطاء اتجاه عصا التحكم أو مفاتيح الأسهم.

STICK (n)

n = 0 استخدام مفاتيح الأسهم
n = 1 استخدام عصا التحكم «1»
n = 2 استخدام عصا التحكم «2»

قف 13 STOP

يستخدم لقطع تنفيذ البرنامج.

STOP

قف نعم / كلا / قف * STOP ON / OFF / STOP

للتعامل أو إبطال مفعول أو الإبقاء على حدث الاعتراض الناتج من ضغط «CTRL + STOP».

STOP ON
STOP STOP
STOP OFF

حزم $ STR$

لتحويل بيان عددي إلى بيان مقطعي.

STRIG — زناد

لتحديد الحالة التي عليها زناد الإطلاق، أو قضيب المسافة.

STRIG (n)

n = 0 استخدام المسطرة كزناد

n = 1, 3 استخدام العصا «1»

n = 2, 4 استخدام العصا «2»

STRIG ON / OFF / STOP — * زناد نعم / كلا / قف

للتعامل مع، أو إبطال مفعول أو الإبقاء على حدث الاعتراض الناجم عن ضغط قضيب المسافة أو زناد الاطلاق لعصا التحكم.

STRIG (n) ON
STRIG (n) OFF
STRIG (n) STOP

n كما في البلاغ «STRIG»

STRING $ — حزمة $

لإعطاء الحرف المناظر لرمز معطى.

```
10    INPUT A
20    PRINT STRING $(A, «X»)
30    GOTO 10
      run
      20
      XXXXXXXXXXXXXXXXXXXXX
```

SWAP — * بادل

استبدال قيمة متغيرين فيما بينهما.

SWAP X, Y

جدول TAB

لتحريك المشيرة في نقلة واحدة عدد من الخانات

```
PRINT TAB (5) B TAB (10) A
      B         A
```

ظا TAN

تعطي ظل الزاوية (نصف قطرية)

```
PRINT TAN (0.5)
.54630248984381
```

وقت TIME

للاحتفاظ أو تعين قيمة المؤقت الداخلي.

لاتتبع TROFF

يوقف عمل البلاغ «TRON».

تتبع TRON

يظهر رقم السطر الذي على وشك التنفيذ على الشاشة.

```
TRON
```

قيمة VAL

تعد الصورة المقلوبة لدالة «STR$» حيث تحول بيان مقطعي إلى بيان عددي.

VAL (مقطع رمزي)

```
PRINT VAL («−100»)       <CR>
      −100
PRINT VAL («GA100»)      <CR>
      0
```

WIDTH 47 عرض

. . . . . .

لتحديد طول السطر في طور النصوص (الكتابة)

SCREEN 0 MAX. 40
SCRREN 1 MAX. 32

هناك بعض الإيعازات الخاصة بالحاسب (NEC) إضافة لتلك التي وردت في الملحق (2). وهذه هي:

CONSOLE

. . . . . .

لإختيار نمط الشاشة

CONSOL [M] [,n] [,f] [,k]

M = خط بداية الدوران للشاشة (0-19)

n = مدى الدوران للشاشة

f = لإظهار وإخفاء أسماء مفاتيح الدوال من أسفل الشاشة (1 للإظهار و 0 للإخفاء).

k = لإظهار وإلغاء الصوت المصاحب للضغط على المفاتيح (1 للإظهار و 0 للإلغاء).

COLOR

. . . . . .

لإختيار لون الكتابة والخلفية والمزج بينهما.

COLOR [f] [,b] [,c]

f = لون الكتابة (0-15)

b = لون الخلفية (0-15)

c = المزج (0-15)

GET [d]

. . . . . .

لحفظ الأشكال الشبحية من على الشاشة.

GET [d] (X1, Y1) – (X2, Y2)

X1, Y1 إحداثي الزاوية العليا اليمنى للشكل

X2, Y2 إحداثي الزاوية السفلى اليسرى للشكل

d الاسم المميز له.

PUT [d]

. . . . . .

لتحميل الأشكال التي تم حفظها باستخدام البلاغ «GET»

PUT [d] (X,Y), d

X, Y إحداثي الزاوية العليا اليمنى للشكل

d الاسم المميز له.

LCOPY

. . . . . .

لنقل ما موجود على الشاشة إلى الطابعة.

LCOPY (F)

F نمط النقل «1» عند استخدام طابعة 300 × 200 نقطة

«2» عند استخدام طابعة 400 × 320 نقطة

LIST L

. . . . . .

لإعطاء أرقام السطور التي لها علاقة مع السطر المبين، والعلاقة تكون بوجود أحد البلاغات: GOTO, GOSUB, ON.. GOTO, ON.. GOSUB.., IF.. ...

LIST L, M

M رقم السطر المطلوب

LIST V

. . . . . .

لإعطاء أرقام السطور التي فيها المتغير المطلوب. وإذا لم يحدد متغير معين، يعطي ذلك لجميع المتغيرات.

LIST V [,d]

d المتغير المطلوب

SCREEN

. . . . . .

لإختيار نمط الشاشة.

SCREEN [m] [,a] [,v]

m نمط العرض (1-4)

a نوع الشاشة (للكتابة أو الرسم)

v الشاشة المعروضة

1 نمط الكتابة (20 × 40)

2 نمط شبه الرسم (40 × 80)

3 نمط الرسم متعدد الألوان والدقة المتوسطة (160 × 200)

4 نمط الرسم ذي الدقة العالية (200 × 320)

ملحق (4) جدول الرموز الانكليزية لـ«MSX صخر»

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		┼		0	@	P		p	Ç	É	á	Ã	▬	◀	α	≡
1	☺	┴	!	1	A	Q	a	q	ü	æ	í	ã	▚	⧗	β	±
2	☻	┬	"	2	B	R	b	r	é	Æ	ó	Ĩ	▞	⧓	Γ	≧
3	♥	┤	#	3	C	S	c	s	â	ô	ú	ĩ	▀	▪	Π	≦
4	◆	├	$	4	D	T	d	t	ä	ö	ñ	Õ	▬	▗	Σ	⌠
5	♣	┼	%	5	E	U	e	u	à	ò	Ñ	õ	▀	▗	σ	⌡
6	♠	├	&	6	F	V	f	v	å	û	ª	Ũ	▌	▄	μ	÷
7	•	—	'	7	G	W	g	w	ç	ù	º	ũ	▟	▚	γ	≈
8	◘	┌	(	8	H	X	h	x	ê	ÿ	¿	IJ	▉	∧	Φ	°
9	○	┐	)	9	I	Y	i	y	ë	Ö	⌐	ij	▊	‡	θ	●
A	◙	└	*	:	J	Z	j	z	è	Ü	¬	¾	▋	ω	Ω	·
B	♂	┘	+	;	K	[	k	{	ï	¢	½	˜	▨	▊	δ	√
C	♀	×	,	<	L	\	l	¦	î	£	¼	◇	▨	▃	∞	n
D	♪	/	-	=	M	]	m	}	ì	¥	¡	‰	▼	▋	φ	2
E	♫	\	.	>	N	^	n	~	Ä	Pt	«	¶	▲	▊	ε	▮
F	☼	+	/	?	O	_	o	△	Å	ƒ	»	§	▶	▃	η	

مثال :

```
CHR$(65) = A
CHR$(&H5A) = Z
```

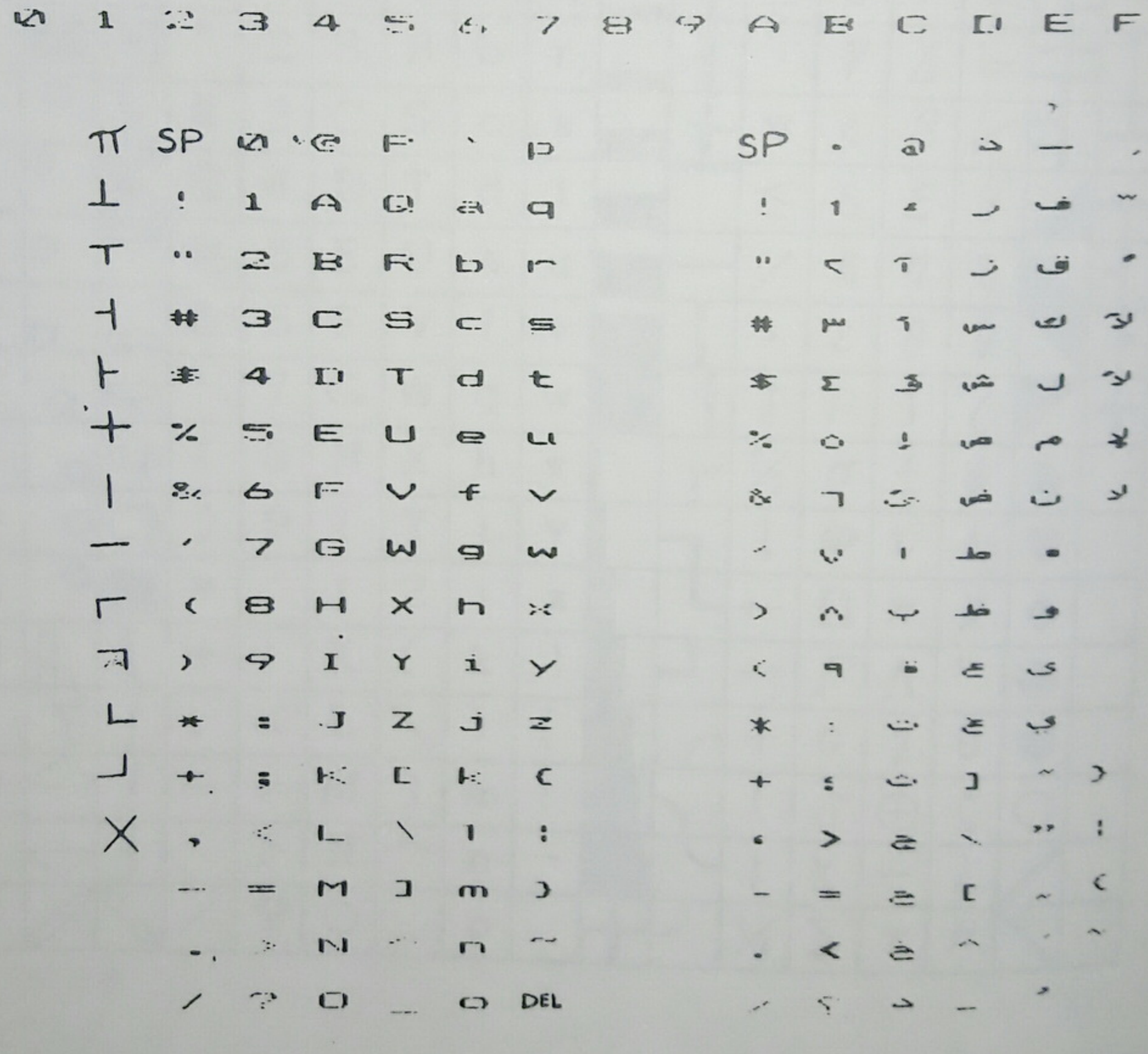
ملحق (5)

جدول الرموز العربية والانكليزية لـ «صخر بيسك» باستخدام كارتردج البيسك العربي

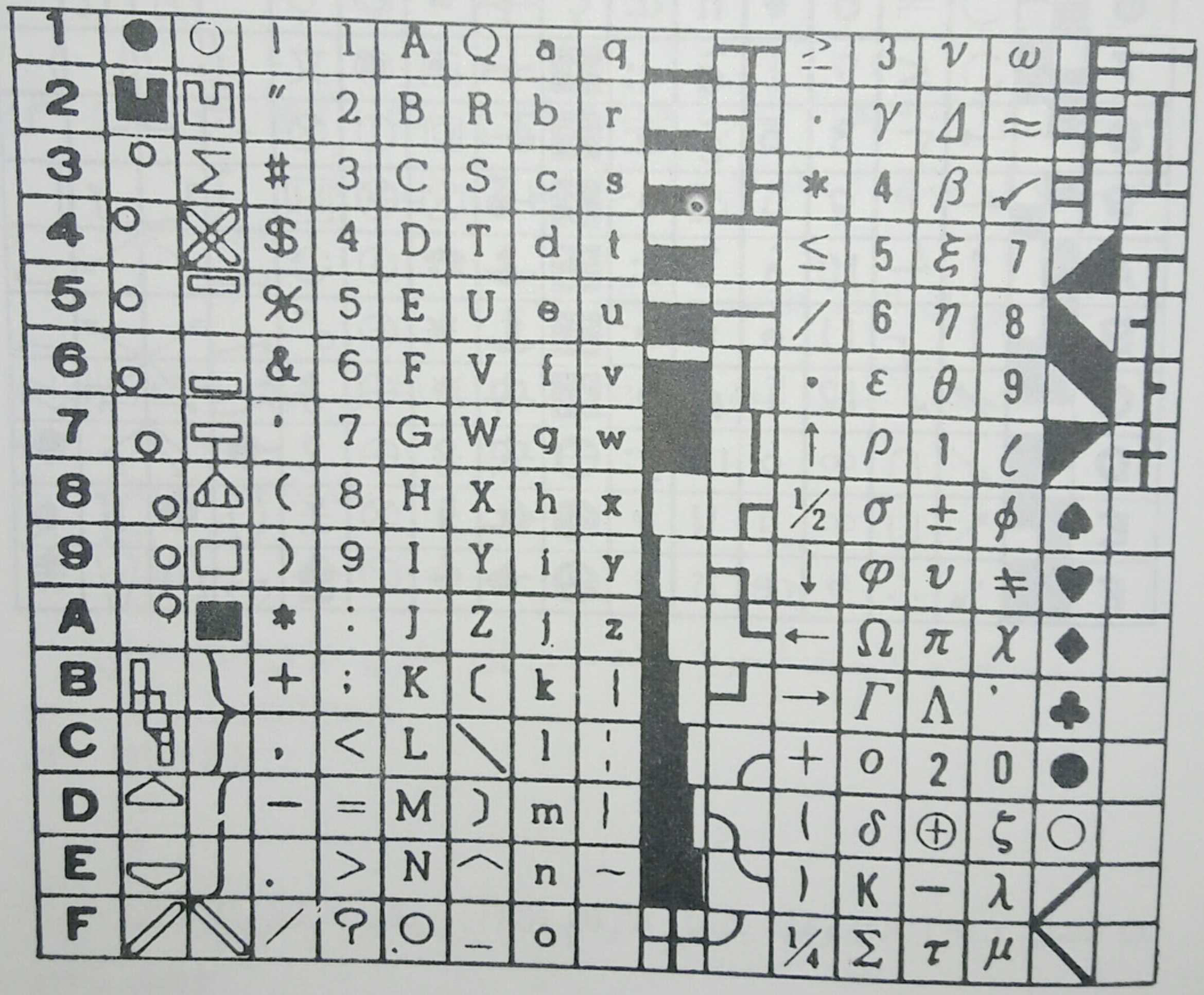
	٠	١	٢	٣	٤	٥	٦	٧	٨	٩	ا	د	ذ	ر	ز	و
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
٠	0		π	SP	.	@	ذ	ـ	ـ			SP	0	@	P	'	p
١	1		⊥	!	١	ه	ر	ـَ				!	1	A	Q	a	q
٢	2		⊤	"	٢	آ	ز	ف	ـً			"	2	B	R	b	r
٣	3		⊣	#	٣	أ	س	ك	لآ			#	3	C	S	c	s
٤	4		⊢	$	٤	ؤ	ش	ل	لأ			$	4	D	T	d	t
٥	5		+	٪	٥	إ	ص	م	لإ			%	5	E	U	e	u
٦	6		∣	&	٦	ئ	ض	ن	لا			&	6	F	V	f	v
٧	7		─	'	٧	ا	ط	ه				'	7	G	W	g	w
٨	8		⌐	)	٨	ب	ظ	و				(	8	H	X	h	x
٩	9		¬	(	٩	ة	ع	ى				)	9	I	Y	i	y
ا	A		└	*		ت	غ	ي				*	:	J	Z	j	z
د	B		┘	+	؛	ث	[	ـُ	]			+	;	K	[	k	{
ذ	C		╳	،	>	ج	\	ـِ	؛			،	<	L	\	l	:
ر	D			-	=	ح	]	ـّ	[			-	=	M	]	m	}
ز	E			.	<	خ	^	ـْ	ـ			.	>	N	^	n	~
و	F			/	؟	د	—	ـٌ	DEL			/	؟	O	—	o	

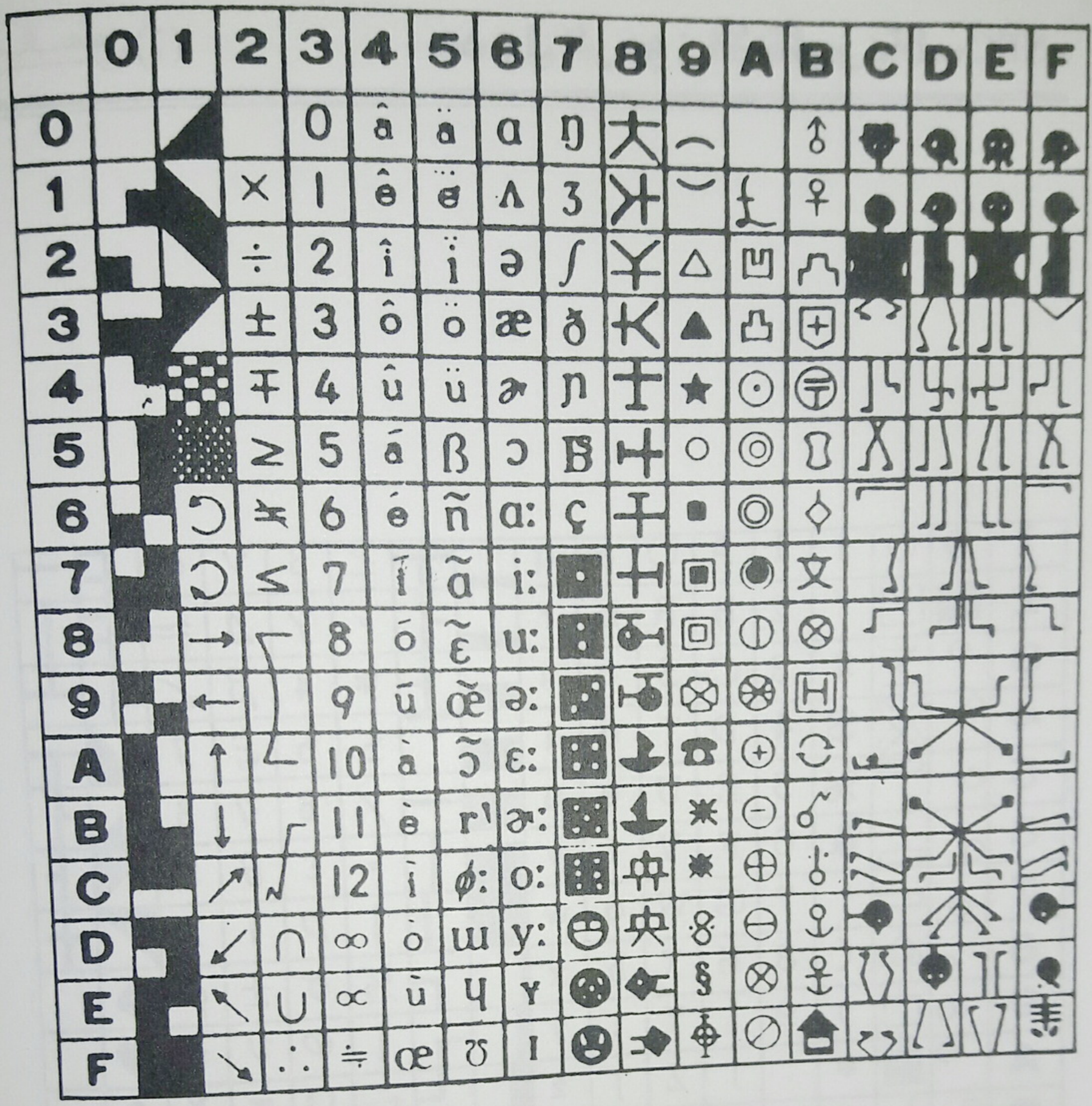
ملحق (6)

جدول الرموز العربية والانكليزية لـ «صخر MSX»



ملحق (7) جدول الرموز الانكليزية لـ«NEC»




0 1 2 3 4 5 6 7 8 9 A B C D E F
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F

ملحق (8) نظام الأرقام

تكتب الأرقام بأكثر من نظام، والنظام المعروف لدينا هو النظام العشري، الذي يمثل الأرقام (0-9) ويتكون الرقم من آحاد وعشرات مئات. والأنظمة الأخرى هي:

□ الثنائي (BINARY)، يتكون هذا النظام من (0-1) فقط، وتستخدم جميع الحاسبات في عملها النظام الثنائي. ويكتب بالشكل التالي:

```
9     = & B 1001
```

&B تدل على أن الرقم المكتوب بالنظام النائي.

ويستخدم البيسك لتحويل عدد عشري إلى الثنائي كما في المثال التالي:

```
10    X = 171
20    PRINT BIN$(X)
      run
      10101011
```

ويستخدم البيسك لتحويل عدد ثنائي إلى عشري كما في المثال التالي:

```
10     X = &B 00101011
20    PRINT X
      run
      43
```

□ الثماني (OCTAL) وهذا يشمل الأرقام (0-7)، ويكتب بالطريقة التالية:

```
477 = &0735
```

يستخدم البيسك لتحويل عدد ثماني إلى عشري كما في المثال التالي:

```
10    X = &0 6517
20    PRINT X
      run
      3407
```

ويستخدم البيسك لتحويل عدد عشري إلى ثماني كما في المثال التالي:

```
10    X = 171
20    PRINT OCT$(X)
      run
      253
```

□ الستعشري (HEXADECIMAL). وهذا يشمل الأرقام (0-9) والأحرف (A-F). يستخدم البيسك لتحويل العدد الستعشري إلى عشري كما في المثال التالي:

```
10    X = &HEFA3
20    PRINT X
      run
      4189
```

&H تدل على أن العدد مكتوب بالنظام الستعشري

ولتحويل العدد العشري إلى ستعشري، لاحظ المثال التالي:

```
10    X = 171
20    PRINT HEX$(X)
      run
      AB
```

الرقم العشري	الرقم الستعشري	الرقم الثنائي	الرقم الثماني
0	0	0	0
1	1	1	1
2	2	10	2
3	3	11	3
4	4	100	4
5	5	101	5
6	6	110	6
7	7	111	7
8	8	1000	10
9	9	1001	11
10	A	1010	12
11	B	1011	13
12	C	1100	14
13	D	1101	15
14	E	1110	16
15	F	1111	17

قائمة المصطلحات

A

ABS	مطلق
AUTO	ذاتي

B

BACKGROUND	خلفية
BASIC (Beginners Allpurpose Symbol Instruction Code)	لغة البرمجة (البيسك)
BINARY	ثنائي
BLACK	أسود (غامق)
BORDER	إطار
BS (BACK SPACE)	الرجوع إلى الخلف

C

CALL	نادي
CLEAR	نظف
CLOSE	اقفل
COLOR (COLOUR)	لون
COMPLETE	اكتمل (انتهى)
COS	جتا
CPU	وحدة الذاكرة
CURSOR	المشيرة

D

DARK	مظلم (غامق)
DATA	بيانات
DEL (DELETE)	الغ (امسح)
DECIMAL	عشري
DIM	بعد
DISPLAY	يعرض
DIVISION	قسمة
DRAW	رسم

E

EDITING	مونتاج
END	نهاية
ERROR	خطأ

F

FILES	ملفات
FIRMWARE	البناء البرامجي الثابت
FORMAT	تنظيم
FOUND	وجد

G

GOSUB	تفرع
GOTO	اقصد
GRAPHICS	رسوم، نقوش
GRAY	رصاصي
GREEN	اخضر

H

HARDWARE	البناء الهيكلي
HEAD	رأس
HEXADECIMAL	ستعشري
HI-RES (HIGH RESOLUTION)	دقة عالية

I

I/O (INPUT/OUTPUT)	دخل / خرج
ILLEGAL	غير مشروع (غير قانوني)
INKEY	كشف
INPUT	ادخل
INS (INSERT)	ادخل (إدخال حرف ضمن كلمة)
INTEGER	عدد صحيح

K

KEY	مفتاح

L

LIST	اسرد
LOAD	حمل
LOCATE	حدد

M

MACHINE CODE	لغة الآلة
MAGENTA	بنفسجي
MEDIUM	وسط (فاتح)
MOTOR	محرك
M.S.X.	العلامة المسجلة لـ (Micro Soft corporation)

N

NEXT	تالي
NUMBER	رقم، عدد

O

O.K	تم، حسناً، مضبوط
OCTAL	ثماني
OPEN	افتح
OVERFLOW	زائد عن الحد

P

PLAY	اعزف
PRINT	اطبع

Q

Q.D (Quick Disk)	القرص السريع

R

RAM	ذاكرة القراءة والكتابة
READ	اقرأ
RED	أحمر
REGOSTOR	سجالة
REM	ملحوظة
RENUM (RENUMBER)	إعادة ترقيم
RESTORE	عاود
RETURN	ارجع
ROM	الذاكرة الثابتة (ذاكرة القراءة فقط)
RUN	نفذ

S

SAVE	احفظ
SCREEN	شاشة
SHIFT	انقل
SHIP	سفينة
SINK	اغرق
SOFTWARE	البناء البرامجي
SOUND	صوت
STATEMENT	شبح (صورة)
STATEMENT	جملة
STOP	قف
SYNTAX	لغوي
SWAP	بادل

T

TEMPO	إيقاع (سرعة العزف)
TRANSPARENT	شفاف
TYPE	نوع

U

UNDEFINED	غير محدد

V

VAL	مقدار (قيمة)
VARIABLES	متغيرات
VDUO (Video Displmay Unit)	وحدة الشاشة

W

WIDTH	عرض

المصـــادر

1 — ANGELL, IAN, O; ADVANCED GRAPHICS WITH THE SINCLAIR ZX SPECTRUM.
2 — AQUARIUS, HOME COMPUTER SYSTEM.
3 — CARTER, GRAHAM, GAMES FOR YOUR MSX.
4 — EIC, PERSONAL COMPUTER, PC.6001BMK II, USERS MANUAL.
5 — EIC, PERSONAL COMPUTERS, PC.6001BMK II, N60M, BASIC MANUAL.
6 — GREGORY, JIM, THE MSX GAMES BOOK.
7 — HERNE, D.H; INTERACTIVE COMPUTING WITH BASIC.
8 — JOSEPH, P.P; COMPUTER PROGRAMMING IN BASIC.
9 — MAUNDER, B; THE ZX 81 COMPUTER.
10 — POOLE, L.; SOME COMMON BASIC PROGRAMMES.
11 — TONY, MARRIOTT; STARTING WITH MSX, KUAMA COMPUTERS LTD.
12 — TONY, WILLIAM; SPECTRUM MACHINE LANGUAGE FOR THE ABSOLUTE BEGINNERS.
13 — TOSHIYKI SATO; THE COMPLETE MSX PROGRAMMERS GUIDE. MELBOURNE HOUSE PUBLISHERS.
14 — YAMAHA, MSX BASIC REFERENCE MANUAL.

15 — العالمية؛ صخر بيسك

16 — العالمية؛ صخر بيسك (دليل كتابة البرامج)

برمجة الحاسبات الشخصية

يعد كتاب برمجة الحاسبات الشخصية من الكتب الفريدة التي تناولت استخدام الحاسب بشكل مبسط، معتمدا اسلوب الرسم والحركة وبرامج الالعاب، ومبتعدا عن الرياضيات المعقدة والمسائل الحسابية.

ويشمل الكتاب عدة فصول ابتداء بطريقة تشغيل الحاسب لتقود القاريء الى طريقة كتابة برامج الالعاب، ويحتوي الكتاب كذلك على عدة ملاحق، تلخص عمل الاوامر المستخدمة بلغة البيسك العامة، مع الاشارة الى الاوامر الخاصة بالحاسبة صخر وكذلك الوركاء NEC بصورة خاصة، ويحتوي كذلك على عدد كبير من البرامج المتنوعة.

مَنْشُورَاتْ وتَوزِيع

المَكْتَبَةُ العَالَمِيَّة

بغداد - شارع السعدون

هاتف ٨٨٨٩٣٥٢

مطبعة الديواني - بغداد

هاتف : ٨٨٧٦١٩٧