البرمجة

خطوة ..... خطوة

على حاسبتي الوركاء وصخر

الدكتور محمد زكي محمد خضر

1 مع ملاحق عن حاسبات اكواريوس ـ سنكلير ـ شارب

هذا الكتاب

هذا الكتاب خطوة لاشاعة استخدام الحاسبات الالكترونية باسلوب بسيط ومتدرج . وهو يخاطب اليافعين كما يخاطب الكبار . لقد اختيرت بعض الحاسبات الالكترونية الشائعة اليوم للبدء بها حيث لا بد لدارس هذا الكتاب ان يكون قريباً من حاسبة الكترونية . فهو يعطي المبادىء الاولية للبرمجة بلغة بيسيك على حاسبتي الوركاء وصخر ويضيف ملاحق عن حاسبات اكواريوس وسنكلير وشارب .

وبالحقيقة يمكن استخدام مادته العلمية مع الغالبية العظمى من الحاسبات الالكترونية الشخصية الاخرى بل وحتى مع الحاسبات الكبيرة مع قليل من التعديلات .

لا يتقيد الكتاب بالاساليب الروتينية لتقديم هذه اللغة بل يبدأ بالاجزاء السهلة الفهم والتي ترفع من قابلية القارىء على البرمجة بسرعة .

يحوي الكتاب المعلومات الاساس للبرمجة بلغة بيسيك والتي اذا ما اتقنها القارىء اتقاناً جيداً استطاع التقدم في استخدام الحاسبات الالكترونية بكفاءة .

حقوق الطبع والنشر محفوظة للمؤلف

السعر ١/٦٠٠ دينار

مديرية دار الكتب للطباعة والنشر
جامعة الموصل

البرمجة
خطوة ..... خطوة
على حاسبتي الوركاء وصخر

الدكتور محمد زكي محمد خضر

مع ملاحق عن حاسبات اكواريوس ـ سنكلير ـ شارب

بسم الله الرحمن الرحيم

اقرأ بأسم ربك الذي خلق
خلق الانسان من علق
اقرأ وربك الاكرم
الذي علم بالقلم
علم الانسان ما لم يعلم

صدق الله العظيم
سورة العلق 1-5

بسم الله الرحمن الرحيم
بين يدي هذا الكتاب ..

الحمد لله الذي علّم بالقلم .. علم الانسان ما لم يعلم ...

فوق هذا الكوكب وفي قديم الزمان .. عاش الانسان .. فتطلع الى معبود يعبده .. ولم يتركه الله تعالىٰ طويلاً في حيرته .. فاخرجه الله تعالى من ظلام الشك الى نور اليقين .. لكنه كان ينسىٰ كلما طال عليه الامد .. وفي تطلعه في الظلام كان يصنع تماثيل تشبه هيأته ليعبدها ويقدسها .. لكن الله علمه بالقلم .. فعرف الكتابة لينقل افكاره الى ما يعقبه من أجيال فيزدادوا علماً .. فكانت الكتابة ذاكرة الانسانية .. واليوم يحاول الانسان أن يصنع نموذجاً اخر من تلك التماثيل .. يصنع ذاكرة مادية يسميها دماغاً الكترونياً ويصنع آلة يسميها انساناً آلياً .. انه يحاول خلق ما يشبهه .. كما فعل الانسان الاول ولكن بهيأة اخرى .. فهل سيفلح الانسان في صنع انسان آلي يصنع ناساً اليين مثله؟ .. والى أي مدىٰ سيسير في هذا الطريق ..؟ قد يفلح الانسان في خطوة بهذا الاتجاه .. وخطوات باتجاه اخر .. لقد حفز التقدم في حقل الحاسبات الالكترونية العلماء على المزيد من البحث في دماغ الانسان للمزيد من المعرفة .. ومحاولين تقليد ما فيه من عجائب خلق الله تعالىٰ .. لقد ادركوا أن دماغ الانسان اعقد مما كانوا يتصورون .. وادركوا أن مالا يعرفونه اكثر بكثير مما عرفوه .. انهم يقفون عاجزين تمام العجز عن اجابة كثير من الاسئلة .. ماسر الحياة في الخلية؟ .. وكيف تخزن المعلومات في خلايا المخ الحية؟ .. وكيف تمحى تلك المعلومات من الخلايا؟ .. وكيف تسترجع؟ .. كيف تتم عملية التذكر؟ .. وكيف يقوم المخ بالعمليات الحسابية والمنطقية؟ .. وكثير كثير غير ذلك من الاسئلة .. لكن الشيء الاكيد الذي نكاد نجزم به أن دماغ الانسان سيستطيع أن يطوّر الحاسبات الالكترونية اكثر فاكثر .. فهو المبتكر الذي سيعلمها كيف تبتكر بوسائلها البدائية التي سنحاول التعرف عليها في هذا الكتاب ..

أن من يتعلم كيف تعمل خلايا ذاكرة الحاسبات الالكترونية يقف مبهوراً امام عمل خلية دماغ الانسان .. ومن يتعامل مع خلية حية يقف مبهوراً امام تركيب جسم الانسان ونظام عمله .. ومن يتعامل مع الانسان يقف مبهوراً امام عجائب الكون الفسيح .. هذا الكون الذي لا يعرف الانسان نهايته ولن يستطيع في مدى عمره أو أعمار أجيال عديدة قادمة أن يغوص فيه بعيداً جداً عن كوكبنا الصغير .. الارض .

وهكذا يجد الانسان أن عليه أن يتواضع لله تعالىٰ اجلالاً لله واعترافاً بصغره هو امـام عظمتـه تعـالى . . . وقلـة علمـه وضعفـه . . « اهم اشد خلقـاً ام من خلقنا » ؟ . . « أأنتم اشد خلقاً ام السماء بناها » ؟ . . وهكذا يعود اكثر علماء اليوم عبقرية وذكاء الى منطق بسيط . . وواضح . . والاثر يدل على المسير . . افلا تدل السماوات والارض على الخالق الواحد الاحد ؟ . .

بلى . . . ونحن على ذلك من الشاهدين . .

وهكذا بأسم الله نبدأ . .

1 . هذا الكتاب

هذا الكتاب خطوة لاشاعة استخدام الحاسبات الالكترونية باسلوب بسيط ومتدرج . وهو يخاطب اليافعين كما يخاطب الكبار . لقد اختيرت بعض الحاسبات الالكترونية الشائعة اليوم للبدء بها حيث لابد لدارس هذا الكتاب أن يكون قريباً من حاسبة الكترونية . فهو يعطي المباديء الاولية للبرمجة بلغة بيسيك على حاسبتي الوركاء وصخر ويضيف ملاحق عن حاسبات اكواريوس وسنكلير وشارب . وبالحقيقة يمكن استخدام مادته العلمية مع الغالبية العظمى من الحاسبات الشخصية الاخرى بل وحتى مع الحاسبات الكبيرة مع قليل من التعديلات .

لقد حاولنا عدم التقيد بالاساليب الروتينية لتقديم هذه اللغة بل قمنا باختيار الاجزاء السهلة الفهم والتي ترفع من قابلية القاريء على البرمجة بسرعة . لذلك فقد تم تأجيل كل التفاصيل المعقدة الى الاجزاء الاخرى من الكتاب .

فالجزء الاول هذا يحوي المعلومات الاساس للبرمجة بلغة بيسيك والتي اذا ما أتقنها القاريء اتقاناً جيداً استطاع التقدم في استخدام الحاسبات الالكترونية بكفاءة أكثر . لذلك ننصح بتطبيق كافة الامثلة وحل التمارين في هذا الكتاب كلها على الحاسبة الالكترونية المتوفرة لديك .

هذا وسيكون المؤلف ممتناً لسماع الملاحظات والتعديلات التي يقترحها القاريء لاخذها بنظر الاعتبار في الاجزاء الاخرى من الكتاب أو في الطبعات اللاحقة أن شاء الله تعالى .

2 . ماهي الحاسبة الالكترونية ؟

جهاز الكتروني يقوم بعمليات حسابية . يتكون من اجزاء مرتبطة مع بعضها . . هناك انواع متعددة من الحاسبات الالكترونية وحجوم متعددة وتتفاوت اسعارها وقابلياتها كثيراً . أبسط هذه الحاسبات هي الحاسبات الشخصية التي شاع استخدامها مؤخراً . تتكون ابسط انواع الحاسبات الشخصية من ما يأتي :

1 . وحدة الادخال : وهي عبارة عن لوحة المفاتيح التي تحتوي على الحروف الابجدية اللاتينية والعربية احياناً وعلى الارقام العشرة ورموز العمليات الاربعة وبعض الرموز الاخرى . يمكن ادخال المعلومات الى الحاسبة الالكترونية بواسطتها .

2 . وحدة الاخراج : هي عبارة عن وسيلة الحصول على المعلومات من الحاسبة الالكترونية . ويمكن أن تكون بشكل جهاز تلفزيون أو شاشة خاصة أو جهاز طابعة باللغة الانكليزية او بالانكليزية والعربية معاً او جهاز رسم .

3 . وحدة المعالجة المركزية : وتحتوي على قلب الحاسبة الذي يحتوي على الذاكرة ووحدة الحساب ووحدة الاتصالات . ويقاس حجم الذاكرة بالكلمات (وتدعى بايت مثلاً بآلاف البايتات ـ تدعى كيلو بايت) .

4 . وحدة الخزن الثانوية : وتخزن فيها معلومات كثيرة بشكل برامج أو بيانات ويمكن أن تكون جهاز تسجيل على شرائط الكاسيت أو جهاز تسجيل على الاقراص المغناطيسية . اما وسيلة الخزن المستخدمة فهي شريط الكاسيت أو القرص المغناطيسي المرن (فلوبي ديسكيت) والقرص المغناطيسي (كويك ديسكيت) .

اذا حصلت على جهاز حاسبة شخصية صخر أو الوركاء فسيكون لديك وحدة ادخال مدمجة مع وحدة المعالجة المركزية . يمكنك استخدام جهاز تلفزيونك للاخراج وجهاز تسجيل الكاسيت أو جهاز القرص المغناطيسي للخزن الثانوي .

سنحاول في الجزء الاول من هذا الكتاب أن نفترض أن لديك الحد الادنى من المكونات وهي وحدة المعالجة المركزية (معها لوحة المفاتيح) ولديك تلفزيون يحوي قناة للترددات الفائقة العلو (UHF أو الموجة 36) .

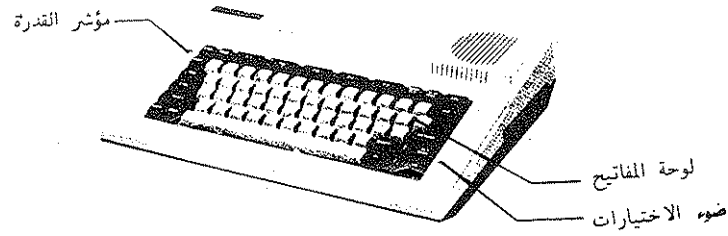
3 . حاسبة الوركاء

انتجت شركة الصناعات الالكترونية في العراق حاسبة شخصية اسمتها الوركاء 6001 بالاتفاق مع شركة NEC اليابانية . وقد احتوى اول نموذج من هذه الحاسبة على استخدام الاحرف اللاتينية فقط . ثم انتجت فيما بعد ذاكرة قراءة ROM للحروف العربية وذاكرة قراءة متنقلة بشكل Cartridge لغرض البرمجة باللغة العربية .

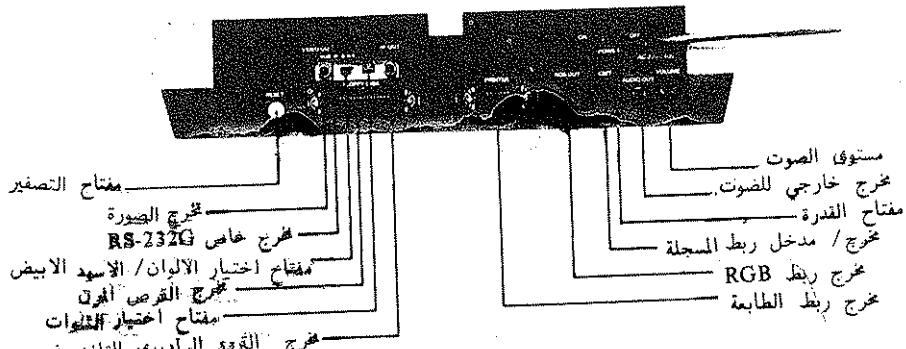
ولغرض البدء بالبرمجة بلغة بيسيك التي تستخدم الحروف اللاتينية سنفترض توفر الجزء الاساس من الحاسبة دون الاضافات العربية على أمل أن نرجع الى استخدام اللغة العربية في الاجزاء اللاحقة من هذا الكتاب أن شاء الله .

يبين الشكل (1) صوراً امامية وجانبية وخلفية لهذه الحاسبة ، كما يبين الشكل (2) كيفية ربط جهاز التلفزيون والمسجلة مع الحاسبة .

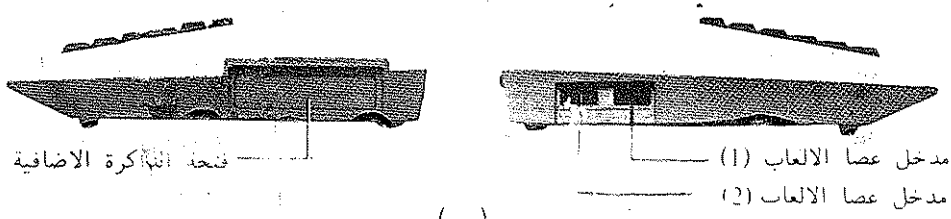
سنفرض بانك اشتريت حاسبة الكترونية من هذا النوع وقمت بنصبها واصبحت الان مستعداً لتشغيل الحاسبة .



(أ)

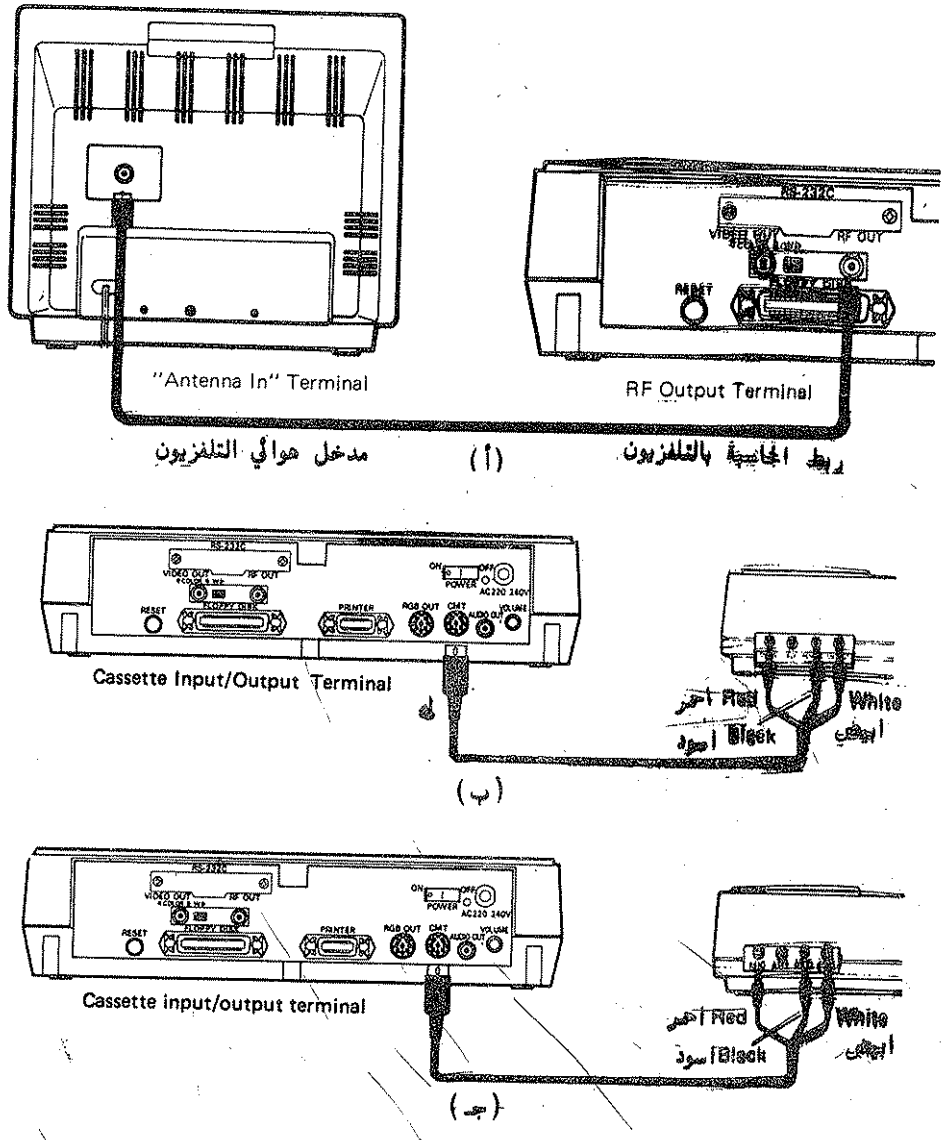


(ب)



(جـ)

الشكل (١) : (أ) لوحة المفاتيح (ب) منظر خلفي (جـ) منظران جانبيان

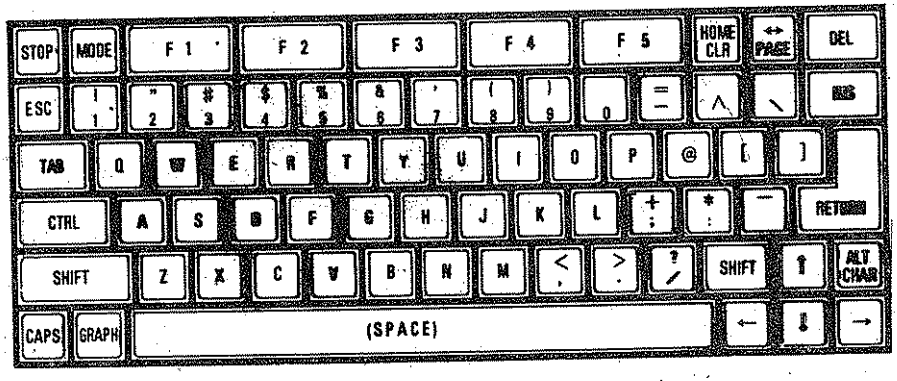


الشكل (2) (أ) ربط جهاز التلفزيون (ب) ربط جهاز التسجيل (جـ) ربط جهاز مسجل البيانات

4 . تشغيل حاسبة الوركاء

تتكون لوحة المفاتيح المبينة في الشكل (3) من ستة صفوف من المفاتيح ويتم تشغيل الحاسبة وفق الخطوات الآتية : ـ

1). افتح مفتاح تشغيل الشاشة التلفزيونية .
2). افتح مفتاح تشغيل الحاسبة الالكترونية .
3). ستجد أن الحاسبة ستظهر خمس اختيارات لمستويات لغة بيسيك المتوفرة عليها . افرض للسهولة اختيار المستوى 5 . اضغط على المفتاح 5 . ثم عدد الصفحات 4
4). ستجد أن الشاشة سيظهر عليها بالانكليزية موديل الحاسبة وعدد الكلمات الفارغة في الذاكرة ثم OK ، وبذلك تكون جاهزة لاستلام ايعازات لغة بيسيك .



الشكل (3) لوحة مفاتيح حاسبة الوركاء

5 . وظائف المفاتيح على حاسبة الوركاء

سنشرح في ما يأتي ملخصاً لوظائف بعض المفاتيح التي تحتاجها في بدء استخدامك للحاسبة ، وسنؤجل المفاتيح الاخرى الى الاجزاء الاخرى من الكتاب أن شاء الله . ولكن قبل ذلك نشير الى أن الموقع الذي سيطبع فيه أول حرف على الشاشة مشار اليه بوسطة بقعة ضوئية مزبعة مستمرة الاضاءة والخفوت . وعند الضغط على أي مفتاح مرة واحدة يطبع الحرف في ذلك الموقع ، وتتحول البقعة الى اليمين ما يكافيء موقع حرف واحد . أما اذا ضغطت على أي مفتاح فترة تزيد

على ثانية واحدة فانك تسمع صوت منبه وتبدأ الحروف المتشابهة بالظهور على الشاشة واذا ما انتهى السطر انتقل المربع المضيء ذاتياً الى السطر التالي وهكذا الى نهاية الصفحة ، فإن انتهت الصفحة زحفت الاسطر كلها سطراً واحداً الى الاعلى (يختفي السطر الاعلى) هذا بالطبع عدا السطر الخاص بالدوال اسفل الصفحة والذي يبقى ثابتاً والذي سنشير الى استخدامه فيما بعد .

عند الضغط على المفاتيح الخاصة بالحروف الابجدية تظهر بصيغتها الصغيرة c a b وهناك ما يشير الى ذلك على الشاشة في سطر الدوال (الزاوية السفلى اليمنى a b c) .

أما اذا اردت الحروف الكبيرة بصيغة A B C فإن ذلك يمكن أن يتم بالضغط على مفتاح SHIFT مع الحرف المعني . ويمكن أن يتم كذلك بالضغط على مفتاح CAPS مرة واحدة فتتحول كافة الحروف الى حروف كبيرة ، وتظهر في كلا الحالتين ABC في سطر الدوال . أما اذا اردت العودة الى الحروف الصغيرة فيتم ذلك بالضغط ثانية على مفتاح CAPS .

يستخدم مفتاحا التزحيف SHIFT الواقعان في اليمين واليسار للحصول على الاختيار الاعلى بالنسبة للمفاتيح المحتوية على أكثر من رمز واحد . ويمكن استخدام أي منهما حسب السهولة .
والمفتاحان GRAPH و ALT CHAR خاصان بالرسم على الشاشة والحروف الرومانية وسوف نؤجل شرحها الآن .
المفتاح الطويل الخاص بالفراغات اسفل لوحة المفاتيح يعطي فراغاً (يحرك البقعة موقعاً واحداً دون أن يظهر أي حرف) .

أما مفتاح الارجاع RETURN فهو لغرض ارسال رزمة من الحروف (مثلاً سطر او جزء منه أو حرف واحد) من لوحة المفاتيح الى الحاسبة الالكترونية لغرض القيام بمعالجتها ومن ثم تعود البقعة المضيئة الى بدء السطر التالي .

واذا كان السطر الذي ارسلته الى الحاسبة ليس ايعازاً مفهوماً وفق قواعد استخدام الحاسبة الالكترونية فستظهر اشارة خطأ

```
? SN Error
OK
```

وبالطبع فإن OK تعني الاستعداد لقبول الايعاز التالي من قبل الحاسبة .

المفاتيح التي تحوي اسهماً (أربعة مفاتيح ↑ ↓ ← →) تقوم بتحريك البقعة الضوئية الى أي موقع على الشاشة حتى وان كان في ذلك الموقع حرف سبق

وان طبع . ويفيد ذلك لتصحيح سطر سبق وان ادخل خطأ ، ولكن اذا ما صحح السطر فيجب الضغط على مفتاح الارجاع RETURN لكي يدخل الى الحاسبة .

هناك مفتاحان في الزاوية العليا اليمنى هما مفتاحا الحذف DEL والحشر INS فالاول عند الضغط عليه يحذف آخر حرف ادخل والمفتاح الثاني عند الضغط عليه (اذا كانت البقعة الضوئية واقعة فوق حرف سبق وان ادخل) فإنه يوفر موقعاً يمكن بواسطته اضافة حرف بين الحروف المدخلة سابقاً . ويفيد ذلك كثيراً لتصحيح اخطاء سابقة .

يستخدم مفتاحا STOP و ESC لغرض ايقاف البرنامج . الاول للايقاف والثاني للايقاف المؤقت ولا يمكن اعادة التنفيذ في الحالة الاولى الا بايعاز خاص ، اما في حالة الايقاف المؤقت فيمكن الاستمرار بالتنفيذ بمجرد الضغط على أي مفتاح كان .

أما مفاتيح الدوال F1 , F2 , F3 , F4 , F5 فتقابل الدوال الخمس الظاهرة على اسفل سطر في الشاشة . فقد ادخل الى هذه الدوال ايعازات محددة هي حسب التسلسل من اليسار color و cload و goto و list و run . ويمكن الحصول على خمس ايعازات اخرى عند الضغط على مفتاح التزحيف SHIFT مع أي من مفاتيح الدوال الخمسة وهي screen و csave و print و play و cont ، وسنؤجل المناقشة حول ذلك الى أن نمر على هذه الايعازات ضمن لغة بيسيك .

أما المفاتيح الاخرى فيندر استخدامها مثل TAB للتحريك بمقدار 8 حروف و HOME CLR لحذف محتويات الشاشة كلها . ومفتاح السيطرة CTRL مفتاح معقد الاستخدام سنؤجل مناقشته الى جزء آخر من هذا الكتاب وكذلك مفتاحي تقليب الصفحات ⟷ PAGE وتغيير الانساق MODE .

اذا ما استخدمت حاسبة الوركاء وظهرت لديك اخطاء فيمكنك الرجوع الى الملحق I .. الذي يحوي ايضاحات عن هذه الاخطاء .

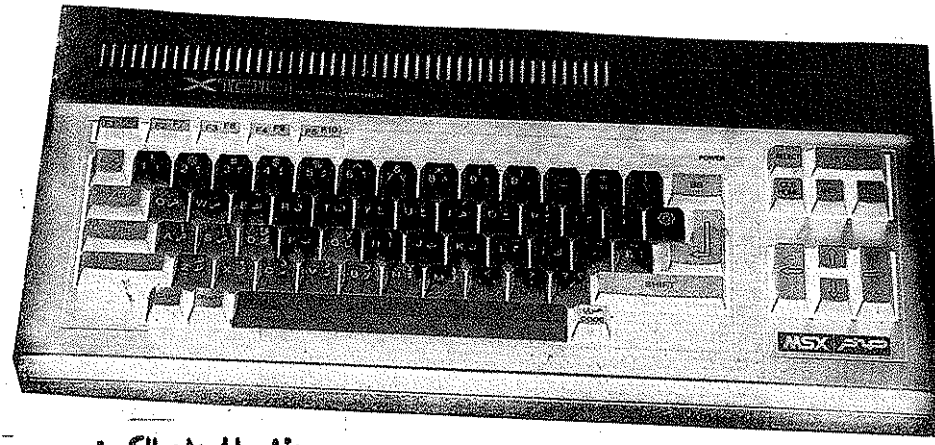
6 . حاسبة صخر : ـ

أنتجت شركة YAMAHA اليابانية حاسبة الكترونية موديل MSX-100 ثم موديل MSX-200 تقبل التعامل مع الحروف العربية وسميت صخر . يحتوي النموذج الاعتيادي على امكانية لاستخدام الحروف العربية ضمن البرامج الانكليزية وهي ميزة ممتازة غير متوفرة حالياً بحاسبة الوركاء . كما أن هناك ذاكرة قراءة متنقلة بشكل Cartridge لغرض البرمجة باللغة العربية ..

ولغرض البدء بالبرمجة بلغة بيسيك التي تستخدم الحروف اللاتينية سنعرض توفر الجزء الاساس من الحاسبة دون الذاكرة الخاصة بالبرمجة باللغة العربية . ويمكن استخدام الحروف العربية ضمن النصوص الانكليزية باستخدام مفتاح العربي ولكن سوف لا نذكر في الجزء الاول من هذا الكتاب امثلة باستخدام اللغة العربية نظراً لعدم توفر ذلك على حاسبة الوركاء ، ولكن المستفيد بامكانه تنفيذ الشرح باللغة التي يشاء طبعاً .

وحيث أن هناك كتيباً ملحقاً بالحاسبة صخر باللغة العربية لذلك سوف لا نخوض في اعادة ما في ذلك الكتيب الصغير من شرح لكيفية التشغيل ووظائف مفاتيح التشغيل والتي تشبه لحد كبير وظائف مفاتيح حاسبة الوركاء التي سبق شرحها .

يبين الشكل (4) حاسبة صخر مبيناً عليها المفاتيح باللغتين العربية واللاتينية .



الشكل (4) حاسبة صخر MSX-200

وسنفترض في هذا الكتاب انك ربطت الحاسبة الالكترونية مع جهاز التلفزيون ذي القناة 36 (موجة فائقة العلو UHF) وادرت مفتاح التلفزيون والحاسبة دون استخدام أي ذاكرة اضافية متنقلة Cartridge فحصلت على عرض بخمس اختيارات والذي باختيارك الرقم 5 تكون قد دخلت الى لغة بيسيك باللغة الانكليزية .

ولغرض اعطائك القابلية على تصحيح الاخطاء التي ربما ستعترضك اثناء تنفيذ برامجك لابد من اعطائك قائمة بالاخطاء التي يمكن أن تشير لها الحاسبة والتي تجدها في نهاية هذا الكتاب في الملحق 2 .

7 . الخطوة الاولى في البرمجة :

الآن وقد امتلكت حاسبة الكترونية شخصية خاصة بك .. سنحاول البدء بالخطوة الاولى ، لنعّرف اولاً ما هو البرنامج .
البرنامج : هو عبارة عن مجموعة من الجمل التي تقوم بتنفيذ مهمة ما على الحاسبة الالكترونية .

فالبرنامج اذاً هو => جملة + جملة + جملة ... الى ن من الجمل .
هذه الجمل ذات علاقة واحدة بالاخرى ، انه كالقصة القصيرة التي تحتوي على عدد من الجمل ذات العلاقة ببعضها البعض .

ولكن .. ما هو اقل عدد من الجمل في البرنامج الواحد ؟ وما أكبر عدد ؟ اقل عدد من الجمل في البرنامج هو جملة واحدة واكبر عدد يمكن أن يكون العشرات أو المئات أو الآلاف حسب حجم ذاكرة الحاسبة الالكترونية التي يتم عليها تشغيل ذلك البرنامج وحسب طبيعة الجمل ايضاً . في الحاسبة التي بين يديك يمكن أن يزيد عدد جمل البرنامج على الالف ، ولكن لا أظن انك ستلاقي صعوبة بهذا الصدد اذا ما حاولت حل مسألة طويلة ، فكثير من المسائل الطويلة لا تحتاج سوى الى عشرات وربما مئات من الجمل .

والان لننتقل الى تعريف الجملة .

الجملة : هي عبارة عن مجموعة من الكلمات والرموز والارقام المرتبطة مع بعضها وفق نسق معين .

ويمكن أن تضم الجملة كلمة واحدة فقط ، ويمكن أن تحتوي على مجموعة معقدة من الرموز والارقام والكلمات كما سنرى .

والكلمات التي تحتويها الجملة معظمها عبارة عن فعل أمر يأمر الحاسبة بفعل شيء مـا والقليل منها للايضاح أو لإكمال غيرها من الكلمات . فهناك بعض الجمل التي يمكن أن تكون مستقلة عن بعضها البعض وهناك جمل يجب أن ترتبط مع بعضها . ويصح الشيىء نفسه مع الكلمات ، فهناك كلمات ضمن جمل ذات علاقة مع بعضهاالبعض وهناك كلمات لا علاقة لبعضها بغيره .

ويمكن أن نميز نوعين من الجمل :

* جمل تنفيذية .
* جمل برمجية .

8 . الجمل التنفيذية والجمل البرمجية :

ماهي الجمل التنفيذية ؟

انها جمل قصيرة تحتوي في غالبيتها على فعل امر واحد يأمر الحاسبة بتنفيذ امر ما يتعلق بالبرنامج ، فهي اذاً جملة خارج البرنامج .

اما الجملة البرمجية فهي جملة من ضمن جمل البرنامج ، وبتعاونها مع غيرها من جمل البرنامج تعطي النتيجة التي يطلبها منها البرنامج بعد تنفيذه باستخدام جمل تنفيذية معينة .

هل هناك فرق بين النوعين من الجمل من ناحية المظهر ؟

نعم . الجملة البرمجية .. يجب أن تبتديء برقم لها يميزها عماسواها الجملة التنفيذية لا تبتديء برقم ، بل تبدأ بفعل الامر مباشرة .

والآن لنبدأ بالتطبيق العملي :

احضر حاسبتك .. شغلها .... ادخل الى لغة بيسيك .. ثم ادخل ما يأتي :

```
10 PRINT "AHMED"
```

هذه جملة برمجية ، لاحظ انها تبتدىء برقم الجملة وهو هنا 10 ، وبعده فراغ أو اكثر . ثم يأتي فعل الامر : اطبع PRINT ، وفعل الامر طبعاً يحتاج الى ما يؤثر به وهنا ماذا نريد أن نطبع ؟ ما نريد أن نطبعه هنا هو كلمة احمد AHMED ولذلك نكتبها بين قوسين مقلوبين .

فهذه جملة برمجية تامة وصحيحة .

ويمكن لهذه الجملة أن تكون عبارة عن برنامج تام وصحيح ، لانها جملة مستقلة لا تحتاج الى اكمالها من قبل جمل اخرى .

اذاً كتبنا الآن برنامجاً تاماً .

ولكن كيف ننفذ هذا البرنامج ؟
انه يحتاج الى جملة تنفيذية هي جملة : نفذ

RUN

هذه الجملة لا تحتاج الى رقم خاص بها ولاتتكون سوى من ثلاثة احرف فقط . ولكن لكثرة حاجتنا لهذه الجملة فإن الحاسبة تساعدنا اكثر من ذلك . فقد خزنت هذه الجملة في احدى الدوال : F_5 ، ولذلك اذا ما اردنا ادخال هذه الجملة فما علينا سوى أن نضغط على المفتاح المؤشر عليه F_5 فقط لا غير . وسنرى أن الجملة (فعل أمر واحد : نفذ) تطبع بكاملها مرة واحدة .

ولكن إذا ما نفذت هذه الجملة بالضغط على مفتاح الارجاع RETURN فانك سترى ما يأتي :
يقوم البرنامج بتنفيذ ما طلب منه ضمن محتوياته من الجمل البرمجية فيكتب كلمة احمد

AHMED

جرّب ذلك . واخبرني . . هل هناك صعوبة ؟

اذا لا قيت صعوبة في ذلك ارجع الى الفقرة الخاصة بحاسبتك في بداية هذا الكتاب واعد قراءتها ثم ابدأ ثانية بهذا الفصل ونفذ البرنامج السابق مرة اخرى . آمل أن لا تجد اية صعوبة أن شاء الله .

لنتقدم خطوة اخرى ، هل تريد أن تستبدل اسم احمد باسمك (اذا لم يكن اسمك احمد فعلاً) ؟ الامر بسيط . بدل كلمة احمد في الجملة المرقمة 10 السابقة اطبع اسمك بالانكليزية . وليس هناك أي مشكلة اذا كان اسمك طويلاً أو قصيراً . كل ما في الامر استمر بادخاله حرفاً حرفاً الى أن تصل الى نهايته ثم ادخل القوسين المقلوبين في نهايته (كما في بدايته) .

واذا ما كان اسمك مركباً يحوي فراغات مثل ABDUL AZIZ ، أو اذا اردت أن تطبع جملة كاملة مثل « اسمي علي »

"MY NAME IS ALI"

فما عليك الا أن تدخل ذلك بتأنٍ حرفاً حرفاً مراعياً الفراغات بين الكلمات جرب ذلك . واعد التشغيل (بادخال جملة التنفيذ RUN بواسطة مفتاح الدالة F_5) .

9 . هيأة جملة البرمجة :

والآن دعنا نقوم بتشريح جملة البرمجة التي نفذناها !!
تتكون هذه الجملة من ثلاثة اجزاء رئيسية هي :

i) رقم الجملة : يتراوح هذا الرقم بين 1 ,65529 . ويمكن أن يكون أي واحد من هذه الارقام .

رقم الجملة هو رقم يميز الجمل عن بعضها لذلك لا يجوز في البرنامج الواحد أن يكون هناك جملتين تحملان الرقم نفسه .

واذا ما اخطأت أنت وكتبت جملة واعطيتها في بدايتها رقماً معيناً ثم ادخلت جملة ثانية واعطيتها الرقم نفسه ، فإن الحاسبة ستظن انك الغيت الجملة السابقة واحللت محلها الجملة الجديدة بالرقم نفسه . لذلك فأنك سوف لن ترى سوى الجملة الثانية . اما الجملة الاولى فانها تكون قد حذفت بمجرد اكمال ادخال الجملة الثانية ، لذلك عليك الحذر الشديد من هذا الخطأ .

ولرقم الجملة فائدة اخرى مهمة جداً وهي أن البرنامج ينفذ الجمل حسب تسلسلها ابتداءً من 1 فما فوق ، الا اذا احتوى البرنامج على اوامر تغير من هذا التسلسل كما سنرى فيما بعد .

فاذا ما ادخلت في برنامج ما الجملتين

```
10 PRINT "AHMED"
5 PRINT "ALI"
```

فإن الحاسبة ستعيد ترتيب هاتين الجملتين ضمن البرنامج بحيث تنفذ الجملة الثانية المرقمة 5 قبل الاولى المرقمة 10 . وستظهر النتيجة على الشاشة

```
ALI
AHMED
```

وهكذا فإن الحاسبة لا تبالي بالتسلسل الذي ادخلت به الجمل المختلفة أنت نفسك بل تسلسل ارقامها التي اعطيتها اياها انت نفسك . وانت حر في اختيار هذه الارقام فيمكن أن تكون الجمل المتعاقبة تحت ارقام 23 ثم 99 ثم 2079 .

ولكن للسهولة يستحسن ترقيم الجمل بالعشرات أي 10 ، 20 ، 30 وهكذا ، وذلك لكي تظهر بمظهر تسلسلي من جهة ولكي يكون لديك مجال فيما بعد اذا غيرت رأيك واردت أن تضع جملة بين الجملتين 10 و 20 فأنك تستطيع أن تضع جملة وتعطيها الرقم 15 مثلاً أو 18 بل ويمكنك أن تضع تسع جمل من 11 لغاية 19 وكل ذلك دون أن تمحي أي جملة أو تغير في أي جملة ضمن البرنامج .

واخيراً فإن رقم الجملة هو بمثابة الاسم لتلك الجملة فاذا ما اردنا أن نرجع الى جملة ما ضمن البرنامج مرة أو مرات عديدة فما علينا الا أن نكتب رقم تلك الجملة ضمن البرنامج فتعلم الحاسبة اننا نعني تلك الجملة لا غيرها ، وسنأتي على امثلة لذلك فيما بعد .

2) فعل الامر ضمن الجملة : كان فعل الامر في مثالنا السابق هو « اطبع » PRINT وهو فعل يحتاج الى ايضاح ما يجب أن يطبع ، ولكن اذا اخطأت ولم تعطي شيئاً لتطبعه الحاسبة بل ادخلت الفعل وحده فإن الحاسبة سوف لا تطبع شيئاً على الاطلاق .

وهناك افعال امر كثيرة سنمر على ذكرها ، وكل واحد منها يحتاج الى نوع مخصوص من المعلومات التي ينفذها .

3) ما يؤمر به : كان في المثال السابق هو كتابة اسمٍ ما مثل "AHMED" فوضعنا الاسم بين قوسين مقلوبين .

ولكن هل ما نريد أن نطبعه يكون دائماً اسماً او كتابة ؟ كلا بالطبع .. فيمكن أن يكون عدداً ، وعند ذلك لاحاجة لوضعه بين قوسين مقلوبين فمثلاً الجملة

```
10 PRINT 5
```

سوف تطبع العدد خمسة .
ولكن هل يكفي ذلك ؟
فربما نريد أن نطبع مقداراً حسابياً مثلاً حاصل جمع خمسة وستة .
هل يمكننا أن نطبع حاصل جمعها ؟ الجواب .. نعم .
وما عليك سوى أن تكتب

```
10 PRINT 5 + 6
```

وعند التنفيذ (الضغط على مفتاح الدالة F_5) سترى أن الناتج سيكون 11 وسيطبع على الشاشة .
اليس هذا اكتشاف خطير ؟ خطير !! خطير !!
هل يمكنني استخدام حاسبتي بمثابة حاسبة جيب لاجراء العمليات الحسابية البسيطة التي تعترضني ؟ الجواب نعم ! نعم ! نعم !
والان جرب الامثلة الاتية :

```
10      PRINT    15 + 24          الجمع
20      PRINT    17 - 5           الطرح
30      PRINT 5 + 27 - 2          جمع وطرح
```

نفذ ، وسترى أن النواتج هي حسب تسلسل ظهورها على الشاشة

```
39
12
30
```

10 . استخدام الحاسبة الالكترونية لاجراء العمليات الحسابية البسيطة

رأينا فيما سبق كيف نستخدم جملة الطبع PRINT لجمع عددين أو الفرق بينهما أو جمع وطرح ثلاثة اعداد .

أن العمليات الحسابية تحتوي : العمليات الاربعة وهي الجمع وعلامتها (+) ، والطرح وعلامتها (-) (حاول أن تميزها عن الشارحة ـ) ، والضرب وعلامتها (*) (وذلك لتمييزها عن حرف اكس X بالانكليزية) ، والقسمة وعلامتها (/) . كما تحتوي العمليات الحسابية احياناً على الرفع الى اسس وعلامتها(^) ،ويمكن أن تحتوي المقادير الحسابية المعقدة على اقواس ( ) .
والان : يمكن ايجاد نتيجة مقدار حسابي معقد دفعة واحدة بواسطة جملة الطبع PRINT وذلك وفق الشروط الآتية :

1) يكتب المقدار اعتباراً من اتجاه اليسار نحو اليمين فمثلاً الجملة 20 في المثال السابق هي 17 مطروحاً منها 5 وليس العكس .

2) اذا احتوى المقدار على اقواس فإنها يجب أن تستخدم بشكل ازواج أي كلما يفتح قوس « (» يجب أن يغلق بعد ذلك «) » . ويمكن أن يحتوي المقدار على أي عدد من ازواج الاقواس . مثلاً

```
(5 + 7) * (4 - 2)
```

3) يعطى للمقادير بين الاقواس الاسبقية في التنفيذ . فمثلاً المقدار اعلاه يعني جمع خمسة وسبعة (الناتج 12) ثم يحسب القوس الثاني : اربعة ناقصاً اثنين (الناتج 2) ثم يضرب حاصلا القوسين أي 12 في 2 ليكون الناتج النهائي هو 24 . وبالطبع نحن سوف لا نشعر بتلك الخطوات ولكن بالناتج النهائي فقط .
بشرط أن نعطي المقدار بالشكل الصحيح .

4) لاحظ أن عملية ضرب مقدارين محصورين بين قوسين يجب أن يكون بينهما علامة ضرب ولايجوز الاستغناء عنها ، حيث أن كتابة المقدار اعلاه بالشكل المعتاد بالجبر

```
(5 + 7) (4 - 2)
```

هو خطأ لاتقبله الحاسبة اطلاقاً ولو جربت ادخاله ونفذت البرنامج فستظهر لك اشارة بأن هناك خطأ معين في هذه الجملة .

5) الاسبقية الثانية التي تقوم الحاسبة بتنفيذها بعد الاقواس هي عملية الرفع فلها الاسبقية على العمليات الاربعة . فمثلاً

```
(4 + 5) ^2 + 3
```

تكون باحتساب ما بين القوسين (أي خمسة زائداً اربعة) ثم يرفع الناتج للاس اثنين فيصبح 81 ثم يجمع له 3 فيصبح الناتج النهائي 84 .

6) الاسبقية الثالثة التي تقوم الحاسبة بتنفيذها هي عمليتا الضرب والقسمة وتنفذان حسب تسلسل ورودها من اليسار الى اليمين فمثلاً المقدار

(4 + 5) / (7 - 4) ^ 2 * 2

يحسب حسب التسلسل الآتي

أ) جمع خمسة واربعة ـ الناتج 9

ب) طرح اربعة من سبعة ـ الناتج 3

جـ) رفع ناتج (ب) أي الثلاثة للاس 2 فيكون الناتج تسعة .

د) قسمة 9 (ناتج أ) على 9 (ناتج جـ) فيكون الناتج 1 .

هـ) ضرب حاصل القسمة د (أي 1) في 2 فيكون الناتج النهائي 2 .

7) بعد تنفيذ الاسبقيات الثلاثة اعلاه (الاقواس ثم الرفع الى الاسس ثم الضرب القسمة) تنفذ عمليتا الجمع والطرح ايهما تسبق الاخرى اعتباراً من اليسار الى اليمين .

لنأخذ مثالاً يحوي مقداراً فيه بعض التعقيد

(7 + 5) ^ 2 / (7 - 5) ^ 3 + (17 - 5) ^ 3 / (7 + 5) ^ 2

هذا المقدار كان قبل كتابته بهذه الصيغة كالاتي وفق الاصطلاحات الجبرية

$$\frac{(7+5)^2}{(7-5)^3} + \frac{(17-5)^3}{(7+5)^2}$$

لذلك فإن خطوات عمل الحاسبة للحصول على الناتج تتم كالاتي :

أ) ايجاد نواتج ما بين الاقواس وهي 5 + 7 = 12 و 7 ناقصاً 5 = 2 وكذلك 17 ناقصاً 5 = 12 ثم جمع مقام الكسر الثاني بجمع 7 + 5 = 12 .

ب) الرفع الى الاسس 12 للاس 2 أي 144 و 2 للاس 3 فتصبح 8 و 12 للاس 3 تصبح 1728 و 12 للاس 2 فتصبح 144 .

جـ) قسمة 144 على 8 = 18 وقسمة 1728 على 144 = 12 .

د) جمع 18 + 12 فيكون الناتج النهائي = 30 .

11 . تمارين

والآن قبل ان نمضي خطوة اخرى نقترح ان تقوم بحل التمارين الآتية بتحويلها الى مقدار حسابي تقبله الحاسبة الالكترونية ثم نفذ ذلك على الحاسبة ويمكنك التأكد من صحة الاجوبة المبينة في نهاية الكتاب :

1) $\left(\frac{14-2}{14+2}\right) \quad \left(\frac{27-3}{9+3}\right)$

2) $$\frac{1}{\left(\frac{27}{9} - \frac{18}{3}\right)\left(\frac{1}{3} - \frac{1}{4}\right)}$$

3) $$\left[\frac{415 \times 32}{830}\right]^2 \div \left[\frac{9 \times 21}{756}\right]^2$$

12. الشرح والحساب معاً

لقد توصلنا فيما سبق الى ان ايعاز الطبع PRINT يمكن ان يستخدم لكتابة كلمات ويمكن ان يستخدم لكتابة الاعداد . والآن يحق لنا ان نسأل السؤال : هل يمكن استخدامه لكليهما معاً؟ الجواب : نعم .

مبدئياً لنحاول استخدام الايعاز نفسه للغرض نفسه مكرراً . مثلاً نريد ان نطبع كلمتين فانه يمكننا ان نعطي كل كلمة على حدة مفصولتين بفارزة مثلاً :
PRINT "AHMED" , "ALI" فهذه الجملة سوف تطبع الاسمين احمد وعلي على السطر نفسه مع فراغ بينهما ، جرب ذلك .
وبشكل مشابه يمكننا ان نجد نتيجتي عمليتين حسابيتين منفصلتين بايعاز واحد مثلاً :

```
20 PRINT  5 + 6 , 7 * 2
```

فان الناتج سيعطي نتيجتي العمليتين على التعاقب من اليسار الى اليمين وهما على التوالي 11 , 14 وبينهما فراغ

واذا ما اردنا ان نطبع ايضاحاً مع النواتج الحسابية فيمكن ان يتم ذلك بمزيج من الشرح والحسابات مع وجود الفوارز بينهما مثلاً

```
25 PRINT "THE RESULT =" , 5 + 17
```

فاذا مانفذ هذا البرنامج باستخدام ايعاز التنفيذ RUN بواسطة المفتاح F_5 فانه سيعطي الناتج الآتي

THE RESULT = 22

فقد كتب العبارة المطلوبة والتي وضعت في البرنامج بين قوسين مقلوبين ثم الناتج من عملية الجمع .

لاحظ ان الايضاح المطبوع في هذه الحالة مفيد في كثير من الاحيان لان الناتج والشرح يظهران متداخلين مما يعطي الانطباع بدقة المسألة ووضوحها . ويمكن ان يتم ذلك بالشكل الذي نرغب به . مثلاً

25 PRINT "5 + 17 =" , 5 + 17

فان ما سيطبع يكون

5 + 17 = 22

حيث ان ماوضع بين القوسين المقلوبين يطبع كما هو دون تحريف أو تغيير سواء كان كتابة شرح أو أرقام أو رموز أو علاقات أو أي شيء سواء كان خطأ أو صواباً فانه يطبع كما هو .

وهكذا نستطيع بهذه الوسيلة طبع أي شيء مطلوب طبعه ضمن الكميات المحسوبة . وتقوم الحاسبة نفسها بترتيب مايطبع على الاسطر . فاذا مازاد عدد الكميات المطلوب طبعها عن اثنين او زاد عدد حروف الشرح عن حد معين قامت باستخدام اكثر من سطر واحد .

جرِّب ان تنفذ مثلاً :

40 PRINT "RESULT1 =", 17+5, "RESULT2 =", 17−5

13. جملة الاحلال الحسابية .

لقد تطرقنا الى ايعاز PRINT في الصفحات القليلة السابقة . والآن سنحاول اخذ الايعاز الثاني وهو ايعاز افرض LET .

يستخدم هذا الايعاز في الجمل البسيطة التي تعطي قيماً للمتغيرات .

تدعى مثل تلك الجمل جمل الاحلال الحسابية . مثلاً

```
10      LET     A = 5
20      LET     B = 7
30      PRINT     A , B
```

الجملة 10 تعطي قيمة 5 للمتغير المسمى A ، والجملة الثانية تعطي قيمة 7 للمتغير المسمى B ، والجملة الثالثة تطبع قيم كل من A و B لذلك فستطبع القيمة 5 ثم 7 .

وهكذا نضيف الى معلوماتنا عدة ملاحظات جديدة هي :

جملة الاحلال البسيط هي عبارة عن جملة تشبه المعادلة طرفها الايسر متغير وطرفها الايمن عدد ويمكن أن يكون مقداراً حسابياً .

جملة الطبع PRINT يمكن أن يكون ما تطبعه عبارة عن اسم لمتغير سبق وان اعطيت قيمته في جملة أو جمل سابقة .

ونود أن نضيف ايضاً أن كلمة افرض LET ضمن جملة الاحلال البسيط هي كلمة اختيارية يمكن حذفها فمثلاً

```
15      A = 5
25      B = 7
35      PRINT "A = " , A , "B =" , B
```

ماذا تلاحظ هنا ؟

أن الجملة 15 جملة صحيحة وهي بالضبط الجملة 10 في المثال اعلاه وكذلك الجملة 25 هي نفسها الجملة 20 السابقة . اما الجملة 35 فهي الجملة 30 نفسها مع اضافة شرح عليها بحيث يظهر الناتج على الشاشة بالشكل

```
A = 5
B = 7
```

وهو شكل واضح حيث تضمن الشرح والنواتج في وقت واحد .

والآن .. هل تستخدم جملة الاحلال فقط بهذا الشكل الذي بيناه ؟

كلا ، فلجملة الاحلال اشكال اخرى . انها الجملة التي يتم بواسطتها ادخال المعادلات والقيم والمقادير الحسابية والجبرية الى الحاسبة الالكترونية .

ولكن علينا استخدامها بقليل من الحذر .

في المعادلتين :

```
x = a + b
y - c = b - x
```

المعادلة الاولى يمكن بسهولة ادخالها الى ما يكافؤها من جمل الاحلال على النحو

```
10 X = A + B
```

اما الجملة الثانية فلا يمكن ذلك مباشرة . لماذا ؟

أن جملة الاحلال في الحاسبة الالكترونية تعني أن المتغير الموجود على يسار اشارة المساواة قد حجز موقعاً داخل ذاكرة الحاسبة الالكترونية . ويتم وفق المعادلة احتساب قيمة الطرف الايمن (جمع قيمتي A و B مثلاً اللتين سبق وان عرفنا قيمتيهما) ووضع ناتج هذا الاحتساب في الموقع المحجوز في الذاكرة والآن لنأخذ المعادلة الثانية . فالطرف الايسر فيها هو y-c ، هل يمكن حجز موقع وتسميته y-c ؟ الجواب كلا ، لان الموقع يجب أن يسمى باسم لمتغير ما . لذلك نحتاج أن ننقل c الى الجهة اليمنى من المعادلة (أي اعادة ترتيب المعادلة) لتصبح

y = c + b − a

وعندها يمكن كتابتها بشكل جملة احلال :

```
20    Y = C + B - A
```

وكل هذا على فرض أن قيم A و B و C سبق وان اعطيت للحاسبة في جمل سابقة .

مثال :

```
5     A = 30
10    B = 17
20    X = (A - B) / (A + B)
30    PRINT  A , B , X
```

هذا برنامج كامل . ادخلنا اليه قيمتي A و B في الجملتين 5 و 10 ، ثم احتسبنا قيمة X التي تساوي $\frac{a-b}{a+b}$ ثم قامت الجملة 30 بطبع كل من A و B والناتج X .

والآن .. ماذا يحدث لو كانت هناك جملتا احلال احداهما تناقض الاخرى ؟ مثلاً :

```
5     A = 7
8     A = 27
```

الجواب أن الحاسبة تنفذ الجملة الاولى اولاً ، ولكن اذا مرت بعد ذلك على جملة جديدة تناقضها فانها تنفذ الجملة الجديدة ايضاً وكأنها تلغي مفعول الجملة الاولى . فإن الجملة 8 أعلاه قد الغت مفعول الجملة 5 حيث أن قيمة A كانت 7 لحد تنفيذ الجملة التالية 8 . ولكن ما أن نفذت الجملة 8 حتى الغيت قيمة A السابقة (7) وحلت محلها القيمة الجديدة (27) .

والآن .. سنسأل سؤالاً غريباً جداً !!

هل يجوز أن نكتب جملة غير صحيحة جبرياً ؟ مثل

```
9     A = A + 1
```

الجواب ، نعم . فهذه الجملة صحيحة في ضوء ما سبق أن شرحناه ، ماذا تفعل الحاسبة حينما ترى مثل هذه الجملة؟ انها لا تدقق أن الجملة صحيحة جبرياً أو غير صحيحة ، لان ليس كل القواعد الجبرية هي قواعد في البرمجة . أن الحاسبة هنا سوف تحسب الطرف الايمن ، تجمع 1) مع قيمة A التي سبق ان ادخلت للحاسبة (اذا كانت الجملة 8 قد ادخلت فقيمتها = 27) . وتجمع لها 1 فتصبح 28 ثم تضع هذه القيمة الجديدة في خزان يدعى A . ولكن كان هناك خزان اسمه A وفق الجملة 8 وكان ما يحويه القيمة 27 ، لذلك فإن القيمة 27 في الخزان A سوف تلغى وتحل محلها القيمة الجديدة 28 وهكذا فإن الجملة 9 هي جملة صحيحة . وليس ذلك فحسب بل انها جملة مفيدة جداً ! فإن هذه الجملة ذات فائدة في العد ، فاذا مر الاحتساب على الجملة 9 لاول مرة فانه يجمع لقيمة A واحداً واذا مر عليها ثانية فانه يجمع لها واحداً آخر .. وهكذا كل ما مر الاحتساب على هذه الجملة اضيف لقيمة A واحداً . ولذلك فإن بإمكاننا أن نعتبر أن قيمة A هي وسيلة لعد مرات عملية معينة فكلما مر الاحتساب عليها زادت واحداً وفي النهاية تكون القيمة المستحصلة هي عدد المرات (على فرض اننا ابتدأنا اولاً بقيمة 1 مثلاً في جملة سابقة مثل :

```
8     A = 1
```

وهكذا بهذه الوسيلة نستطيع أن نوسع استخدام جملة الاحلال الحسابية لا لأحتساب المقادير الحسابية والجبرية ولكن كوسيلة للعد .

14. أمثلة

مثال -1 : اكتب برنامجاً يحول الدرجات الفهرنهايتية الى درجات مئوية لقياس درجات الحرارة . افرض أن المراد تحويل درجة الحرارة 50 فهرنهايت الى مئوية .

الحل : لنفرض أن F هي درجة الحرارة بالفهرنهايت لذلك :

```
10    F = 50
```

وإن العلاقة بين درجة الحرارة المئوية م (C) والفهرنهايتية ف (F) هي

$$م = \frac{5}{9} \times (ف - 32)$$

```
20    C = (F - 32) * 5 / 9
```

ونطبع الناتج بعد ذلك

```
30    PRINT "F =", F , "C =" , C
```

حاول أن تنفذ هذا البرنامج المكون من هذه الجمل الثلاث .

مثال -2 : احسب مساحة مثلث اطوال اضلاعه 3 ، 4 ، 5 وفق العلاقتين

م = $\sqrt{\text{ح (ح − أ) (ح − ب) (ح − جـ)}}$

حيث ح = نصف المحيط = $\frac{\text{أ + ب + جـ}}{2}$

الحل : لنفرض أن الاضلاع هي A , B , C وان نصف المحيط هو S والمساحة هي R

```
10      A = 3
20      B = 4
30      C = 5
40      S = (A + B + C) / 2
50      R = (S * (S-A) * (S-B) * (S-C)) ^ 0.5
60      PRINT R
```

لاحظ أننا استخدمنا الرفع الى الاس نصف 0.5 بدل أخذ الجذر التربيعي للمقدار كما استخدمنا اقواساً بعضها داخل بعض .

مثال -3 : راتبك 120 ديناراً وقد مضى من الشهر 23 يوماً . وقد صرفت لحد اليوم 103 ديناراً . هل كان ماصرفته متناسباً مع أيام الشهر الكلية ونسبة ماصرفت الى راتبك الكلي ؟

الحل : لايجاد النسبة المئوية لما مضى من الشهر (23) الى كل أيام الشهر (30) يوماً نقسم 23 على 30 ونضربه في 100% ، وكذلك الحال بالنسبة لايجاد النسبة لما صرفت من راتبك .

لنفرض أن نسبة مامضى من الشهر هي D ونسبة ماصرفت هي S . لذلك

```
10      D = 23 / 130 * 100
20      S = 103 / 120 * 100
30      PRINT "PERCENTAGE DAYS = " D,
        "PERCENTAGE DINARS =" , S
```

15. تمارين

1. اكتب برنامجاً يستلم قياسات الاطوال بالاميال ويعطي الناتج بالكيلو مترات .
2. اكتب برنامجاً يقوم بحساب حجم مخروط بعد معرفة الارتفاع ونصف قطر القاعدة .
3. أكتب برنامجاً يقوم بادخال وارد شخص معين وحساب مايجب أن يدفع من ضرائب يبلغ مجموعها 10٪ من وارده ومصاريف ثابتة مقدارها 15 دينار شهرياً .احسب كل فقرة من هذه الفقرات على حدة والدخل الصافي للشخص .

16. ايعاز الادخال

ماذا تعلمت في البرمجة لحد الان؟

لقد تعلمت الكثير ... ولكن لاتعلم ...

لقد عرفت كيف تستخدم الحاسبة كحاسبة بسيطة ، وكيف تحسب المقادير الحسابية والجبرية ، وكيف تعوض قيماً في معادلات وكيف تطبع النتائج ، وكيف تطبع ايضاح النتائج وكيف تستخدم معداداً ضمن البرنامج .. وغير ذلك من تفاصيل اخرى .

والان .. دعنا نتعلم ايعازاً جديداً ، وهو ايعاز الادخال INPUT . لنرجع الى المثال 1 في الفقرة 14 . لقد قمنا بتحويل درجة الحرارة 50 من فهرنهايت الى مايقابلها من درجات مئوية . ولكن لو اردنا تحويل درجة حرارة اخرى مثل 140 فماذا نفعل؟

الجواب : بالطبع يجب أن نقوم بتغيير الجملة 10 لكي تصبح

```
10 F = 140
```

ولو اردنا ان نحول درجة حرارة مقدارها 80 فهرنهايت الى مئوية ، فماذا نفعل؟ الجواب اننا نغيرها مرة ثانية .. وهكذا .

ورب سائل : هل هذا اسلوب مناسب؟ اي كلما اردنا ادخال قيمة جديدة نقوم بتغيير البرنامج لكي يناسب تلك القيمة؟ الجواب : كلا . هذا ليس الاسلوب الامثل لذلك . فإن البرنامج يفترض فيه ان لايتغير وفق القيم المختلفة لانه عبارة عن صياغة لاسلوب حل مسألة ما . وهذا الاسلوب لايتغير اذا تغيرت قيمة متغير ماضمن المسألة بل يبقى كما هو . وكل مايتغير هو قيمة ذلك المتغير فحسب .

وهكذا فإن جملة أدخل INPUT تساعدنا على التغلب على هذه العقبة فإن اصل البرنامج لايشترط ان ندخل فيه قيمة درجة الحرارة الفهرنهايتية المطلوب تحويلها الى مئوية بل يكفي ان نشير الى تسميتها مبدئياً وندخل قيمتها فيما بعد . ان صيغة ايعاز الادخال INPUT هي

```
10 INPUT F
20 C = (F-32) * 5 / 9
30 PRINT C
```

هذا البرنامج يقوم باستقبال قيمة درجات الحرارة بالفهرنهايت ويحسب ما يقابلها بالدرجات المئوية ثم يطبع الناتج واظنك ستسأل فوراً ، ولكن كيف سيعرف ان قيمة درجة الحرارة بالفهرنهايت هي 50 وليس 140 مثلاً ؟

مهلاً .. مهلاً .. لاتعجل علينا .. وانظرنا نخبرك اليقينا .. جرب ان تشغّل هذا البرنامج بادخال ايعاز RUN او الضغط على المفتاح F_5 . سترى ان الحاسبة ستظهر على الشاشة علامة استفهام ؟ فهي بذلك تستفسر عن قيمة المتغير F الذي يجب ان تدخل قيمته (ادخل القيمة 50 ثم اضغط على مفتاح الارجاع RETURN) وعند ذلك ستحصل على الناتج كما يجب (الدرجة المئوية المكافئة لـ 50 فهرنهايت هي 10 مئوية) .
على ماذا يدل ذلك ؟

بواسطة ايعاز الادخال INPUT يمكننا فصل البرامج عن البيانات . بحيث اذا تغيرت البيانات يبقى البرنامج كما هو دون تغيير .

والان .. جرب أن تشغل البرنامج السابق مرة اخرى بالضغط على مفتاح F_5 فانك سترى علامة الاستفهام مرة اخرى . ادخل درجة الحرارة الجديدة التي ترغب بتحويلها من فهرنهايت الى مئوي ولتكن 140 سترى ان الناتج الجديد (60) سيظهر على الشاشة . وهكذا فانك تستطيع الان ان تحول اية درجة فهرنهايتية الى مئوية بالبرنامج نفسه دون أن تغير فيه شيئاً . كل ماتفعله هو أن تدخل الدرجة الجديدة المطلوب تحويلها . ورب سائل يسأل ! هل يمكن أن نوضح المطلوب ببعض الشرح ، لكي ندخل القراءة الفهرنهايتية ؟ ولم لا ! اضف جملة اطبع قبل جملة الادخال مثلا :

```
5 PRINT "GIVE THE TEMPERATURE IN FEHRENHITE"
```

وهكذا فان هذه الجملة (اعط درجة الحرارة بالفهرنهايتية) سوف تطبع ثم تظهر علامة الاستفهام بعدها عند تنفيذ البرنامج ، جرب ذلك .

وماذا يحدث لو كان في البرنامج اكثر من متغير واحد يطلب اعطاء البيانات له؟ مثلاً المثال -2 في الفصل السابق لايجاد مساحة مثلث عرفت اضلاعه الثلاثة؟ الجواب : ان مايصلح لمتغير واحد يصلح لثلاث متغيرات . فبدلاً من الجمل 10 , 20 , 30 التي ادخلنا بواسطتها قيم الاضلاع A , B , C يمكننا الاستعاضة عنها بجملة واحدة هي

25 INPUT A , B , C

عند التنفيذ يجب أن تتذكر ان عليك ادخال ثلاث قيم تمثل اطوال اضلاع المثلث الثلاثة . وتفصل بينها بالفارزة (,) وذلك حينما تظهر علامة الاستفهام؟ طلباً للبيانات « اي 3,4,5 للمثلث السابق ذكره » .
واذا ما أردت مساحة مثلث اخر ، فما عليك الا أن تعيد التنفيذ (F9) ، وتدخل الاطوال الجديدة .
ولكن ماذا يحدث لو انك ادخلت ضلعين ونسيت الضلع الثالث؟
سوف تظهر على الشاشة علامتان للاستفهام (؟؟) تطلب المزيد ، اي الضلع الثالث . وعند ادخاله يتم اعطاء الجواب .

جرب أن تدخل ضلعاً واحداً اولاً فقط ، سترى ظهور علامتي الاستفهام . واذا ما أدخلت الضلع الثاني فسترى ظهورها مرة ثانية طلباً للضلع الثالث . وعند ادخال الضلع الثالث تظهر النتيجة . أليس ذلك ذكاءً؟ ولكن من الذي علّمها الذكاء؟ ومن الذي علّم الذي علّمها ؟ .

والان .. ماذا سيحدث لو انك ادخلت اربعة اضلاع للمثلث بشكل قيم بينها فوارز؟ ان الحاسبة ستأخذ القيم الثلاث الاولى وسوف تهمل القيمة الاخيرة .

17. المتغيرات الحرفية

ان كافة المسائل والامثلة التي مرت بنا لحد الان كانت المتغيرات فيها اعداداً او ارقاماً سواء في الادخال او في الاخراج . ولكن هل يمكن التعامل مع اسماء الاشخاص مثلاً بالطريقة نفسها؟ مثلاً ان تسأل الحاسبة عن اسم الشخص ضمن البرنامج ثم تقوم بكتابته . الجواب ، نعم .

هناك نوع من المتغيرات يمكن تمييزه عن المتغيرات التي تأخذ قيماً عددية لنرجع الى المثال -3 في الفقرة 14 .

نريد ان ندخل اسمك قبل حل المسألة ثم ندخل الراتب وعدد الايام التي مضت من الشهر بجملة واحدة : كيف يتم ذلك؟

لنفرض ان المتغير الحرفي الذي سندخل الاسم فيه هو N$ وان ماصرفت لحد اليوم هو A وعدد ايام الشهر الماضية (تاريخ اليوم) هو B .

وعند ذلك يمكن كتابة ذلك على النحو الآتي

```
5  PRINT "WHAT IS YOUR NAME?, SPENDINGS AND DATE"
8     INPUT N$, A, B
10    D = B / 30 * 100
20    S = A / 120 * 100
30    PRINT N$ ,"":YOU HAVE SPENT TILL TODAY,"
        S, "PERCENT OF YOUR SALARY IN", D,
        "PERCENT OF THE MONTH".
```

والآن .. اليك النتيجة :

على فرض أن راتبك 120 ديناراً شهرياً (ثابت ضمن البرنامج) وإن اسمك هو احمد وأنك قد صرفت 48 ديناراً في الايام التسعة الاولى من الشهر فإن المدخلات ستكون بعد أن تظهر لك العبارة الاولى على الشاشة :

WHAT IS YOUR NAME?, SPENDINGS AND DATE?

فانك ستجيب

AHMED, 48, 9

وعند الضغط على المفتاح F_5 ستظهر لك النتيجة :

AHMED: YOU HAVE SPENT TILL TODAY
40 PERCENT OF YOUR SALARY IN 30
PERCENT OF THE MONTH.

وهكذا فإن الحاسبة قد اعلمتك انك قد انفقت أكثر مما يجب وعليك الاقتصاد فيما تبقى من الشهر فإنك قد صرفت 40 ٪ من راتبك في 30 ٪ من ايام الشهر ويعني ذلك إنه بقي لك 60 ٪ من الراتب لتصرفه في 70 ٪ من أيام الشهر والا اضطررت الى الاستدانة قبل نهاية الشهر ، ولا تبذر تبذيراً .. إن المبذرين كانوا اخوان الشياطين .. وكان الشيطان لربه كفورا .

لاحظ ان اسمك "AHMED" قد ظهر ضمن الشرح وكأنه جزء من اجابة الحاسبة .فهي الآن قد تعرفت عليك شخصياً وتجيبك باسمك الصريح .

ماذا نستنتج اذاً ؟ إن الاسماء يمكن ان تدخل ضمن عملية الادخال ويعطى لها متغيراً باسم تقع في نهايته علامة العملة $. ويمكن في اي وقت طبع ذلك الاسم

ضمن الشرح المطلوب أو بدون شرح ، مع غيره من المتغيرات العددية أو بدونها حسب مايروق لنا .

ونود أن نشير هنا الى ملاحظة عابرة ... هي أن الراتب الشهري الذي كان 120 ديناراً قد وضع ضمن السؤال وادخل ضمن الحل دون أن نغيره من مسألة الى مسألة عند اختلاف المصاريف وبمضي ايام اخرى من الشهر . وذلك على اساس إنه ثابت لشخص معين ، لذلك فهو ليس من البيانات المتغيرة التي نضطر لتغييرها من يوم لآخر . اما اذا كان هناك شخص آخر ذي راتب مختلف يريد أن يستخدم البرنامج نفسه فعليه حينئذ ان يدخل راتبه ضمن البرنامج . وهكذا يجب ادخال البيانات الثابتة ضمن البرنامج نفسه أما البيانات المؤقتة او ذات الطبيعة المتغيرة فتوضع خارج البرنامج وتعطى اثناء تنفيذ البرنامج وذلك باستخدام ايعاز الادخال INPUT .

كما ان ادخال المتغيرات الحرفية مفيد لادخال وطبع الاسماء وغيرها من كلمات مفيدة .

18. جملة تنفيذية اخرى : اعرض أو سطرٌ LIST

سبق ان أخذنا جملة تنفيذ لتشغيل البرنامج : نفذ RUN والتي وجدناها مخزونة في الدالة F_5 . سنأخذ الان جملة اخرى هي جملة اعرض LIST ، هذه الجملة مخزونة هي الاخرى في الدالة F_4 .

تقوم هذه الجملة عند ادخالها حرفاً حرفاً او بالضغط على المفتاح F_4 بادراج او تسطير أو عرض البرنامج الموجود في الحاسبة والذي سبق ادخاله ، يتم ذلك بشكل متسلسل حسب ارقام الجمل تصاعدياً . وبالطبع فإن ذلك مفيد لاعادة النظر في بعض الجمل او تصحيحها او اضافة جملة جديدة بين جملتين أو في آخر البرنامج .. الى غير ذلك من الفوائد .

ونود هنا أن نشير ايضاً الى انه ليس من الضروري عرض كل البرنامج بل يمكن عرض بعضه فقط شرط تحديد ارقام الجمل التي نريد عرضها . وليس هذا فحسب بل يمكن عرض جملة واحدة فقط .

فمثلاً

LIST 27

تعني إعرض الجملة المرقمة 27

و

LIST 10 – 30

تعني طلب عرض الجمل اعتباراً من الجملة المرقمة 10 لغاية نهاية الجملة المرقمة 30 وستعرض كافة الجمل المرقمة بين هذين الرقمين .
والايعاز LIST 100-

يعني عرض الجمل اعتباراً من الجملة المرقمة 100 (داخلة) لغاية نهاية البرنامج ، كما ان بالامكان عرض البرنامج من بدايته الى جملة معينة مثلاً
LIST -- 100

فذلك يعني عرض الجزء من اول جملة في البرنامج لغاية نهاية الجملة المرقمة 100 . وتجدر الاشارة هنا الى ان الجملة التنفيذية السابقة RUN كان بالامكان استخدامها بشكل مشابه مثلاً

RUN 100

وذلك يعني نفذ البرنامج اعتباراً من الجملة 100 لغاية نهاية البرنامج . وليس هناك امكانية لتحديد نهاية التنفيذ اي ان هناك صيغتان لجملة نفذ (بدون اي رقم : لتنفيذ البرنامج كله او برقم واحد فقط : لتنفيذ جزء من البرنامج اعتباراً من الجملة المحددة بذلك الرقم الى نهاية البرنامج) .

وبالطبع فانه ما ان يبدأ البرنامج بتنفيذ احد هذين الايعازين (RUN أو LIST) فليس بالامكان ايقافه الا بواسطة الضغط على مفتاح الايقاف STOP وعند ذاك تتوقف العملية التنفيذية .

19. تغيير تسلسل تنفيذ البرنامج : جملة اقصد GOTO

قلنا بان تسلسل تنفيذ البرنامج يجري بالضبط وفق تسلسل ارقام الجمل الواردة فيه اعتباراً من اصغر رقم تصاعدياً ، لكن ذلك في كثير من الاحيان لايفي باحتياجات المبرمج بل يريد ان يتحكم بتسلسل التنفيذ بشكل أو بآخر .

هناك عدة جمل تقوم بتغيير تسلسل التنفيذ احدها هو جملة اقصد GOTO والتي تقوم بنقل التسلسل عند وصوله الى الجملة التي تشير اليها هذه الجملة .

لنفرض ان لديك عدداً كبيراً من قراءات درجة الحرارة بالقياس الفهرنهايتي وتريد تحويلها الى قراءات مئوية . امعن النظر في البرنامج الآتي :

```
10  INPUT F
20  C = (F-32) * 5 / 9
30  INPUT C
40  GO TO 10
```

ان هذا البرنامج سيقوم عند تنفيذه (باستخدام F_5) بطلب قيمة المتغير F بواسطة علامة الاستفهام ؟ وعند ادخال قيمته والضغط على مفتاح الارجاع يقوم بتنفيذ الجملة 20 باحتساب درجة الحرارة بالقياس المئوي ثم ينفذ الجملة 30 بطبع قيمة C وهي الدرجة بالقياس المئوي ثم يصل الجملة 40 التي تخبره بالذهاب فوراً الى الجملة المرقمة 10 والتي ستطلب درجة حرارة جديدة بالقياس الفهرنهايتي حيث ستظهر علامة الاستفهام؟ فاذا ما ادخلت درجة حرارة جديدة فان الناتج سيظهر على الشاشة وفق الجملة 30 ثم يعود مرة ثالثة الى الجملة 10 لطلب درجة حرارة جديدة .. وهكذا يستمر طلب درجات حرارة ليحولها من القياس الفهرنهايتي الى المئوي مرة بعد اخرى ويقوم بطبع الناتج كل مرة .

واذا ما اردت ايقاف البرنامج فما عليك الا الضغط على مفتاح الايقاف STOP وعند ذلك يتوقف البرنامج .

لنحاول تطوير هذا البرنامج قليلاً ، لنفرض اننا لانكتفي بطبع درجة الحرارة المئوية بل نريد ان نعرف تسلسلها ضمن قراءات درجة الحرارة التي ادخلناها الى الحاسبة . لذلك فنحن بحاجة الى وسيلة لعد القراءات أو بكلمة اخرى استحداث معداد . تذكر اننا ذكرنا في الفقرة 13 ان أحدى فوائد جملة الاحلال الحسابية هي استخدامها كمعداد . لنفرض اننا سنستخدم المتغير I كمعداد .

```
5  I = 1
10 INPUT F
20 C = (F-32) * 5/9
30 PRINT I,C
35 I = I + 1
40 GOTO 10
```

لقد اضفنا للبرنامج السابق هنا جملتين مختلفتين فقط هما 5 و 35 الجملة 5 ابتدأت البرنامج باعطاء قيمة 1 للمتغير I . أي انها بكلمة اخرى اعطت تسلسل القراءة الاولى وهو التسلسل الاول . الجملة 35 زادت التسلسل بواحد أي اصبح لاول مرة : اثنين وعند العودة الى الجملة 10 من خلال جملة اقصد المرقمة 40 فانه يستلم القراءة الثانية ثم الثالثة ... الخ .

لاحظ اننا في الجملة 30 أضفنا طلب طبع قيمة I مع درجة الحرارة المئوية C . وحيث اننا بينا أن I يمثل تسلسل درجة الحرارة المطلوب تحويلها ، لذلك فإن هذه الجملة سوف تطبع لنا التسلسل ودرجة الحرارة كما هو مطلوب .

20 . تمارين

1 . اكتب برنامجاً يقوم بكتابة مضاعفات السبعة على الشاشة .
2 . اكتب برنامجاً يقوم بكتابة السنوات الكبيسة اعتباراً من بداية القرن العشرين .
3 . اكتب برنامجاً يكتب جدول تحويل الانجات الى سنتمترات حيث أن الانج = 2.54 سنتمتر .

21 . جملتا القراءة READ والبيانات DATA

سبق أن شرحنا وسيلة ادخال البيانات الى الحاسبة باستخدام جملة الادخال INPUT ، ولكن تلك الجملة تحتاج الى ادخال البيانات عند تنفيذ البرنامج قراءة قراءة . هناك وسيلة لادخال كافة البيانات التي نحتاج ادخالها دفعة واحدة وذلك باستخدام جملة اقرأ READ .

أن صيغة هذه الجملة تشبه جملة ادخل INPUT السابقة في أن المتغير المطلوب ادخال بياناته يعقب تلك الجملة . لكن هذه الجملة تحتاج الى جملة مصاحبة لها (ضمن البرنامج نفسه) للبيانات DATA . ويمكن أن يعطى في جملة بيانات واحدة الارقام التي تحتاجها جملة READ واحدة أو اكثر

لنفرض اننا نريد احتساب درجات الحرارة المئوية المقابلة للدرجات الفهرنهايتية 14 و 23 و 32 و 41 و 50 فقط وذلك باستخدام جملتي اقرأ والبيانات . يمكننا لهذا الغرض تحوير البرنامج الذي سبق لنا شرحه في الفقرة السابقة على النحو الآتي :

```
5     I = 1
11    READ    F
20    C = (F - 32) * 5 / 9
30    PRINT   I,C
35    I = I + 1
40    GO TO   11
50    DATA 14 و 23 و 32 و 41 و 50
```

وهكذا ترى أن البرنامج بقي نفسه ، فقد استعضنا عن جملة الادخال 10 بجملة القراءة المرقمة 11 بالاضافة الى جملة البيانات 50 والتي تحتوي على كافة درجات الحرارة المطلوب تحويلها من الفهرنهايتية الى المئوية .

وسترى أن الجواب سيكون كالآتي (بعد تنفيذ البرنامج بالضغط على مفتاح الدالة F_5)

```
1        -10
2         -5
3          0
4          5
5         10
OUT OF DATA IN 11          في حالة الحاسبة صخر
(؟OD Error in 11 في حالة الوركاء)
```

من الواضح ان الحقل الاول كان تسلسل درجة الحرارة والحقل الثاني هو درجة الحرارة بالتقدير المئوي .

ولكن الملفت للنظر هو الملاحظة الاخيرة بعد انتهاء طبع كافة القراءات الخمس ، هذه الملاحظة هي نتيجة طبيعية لجملة اقصد GΘTΘ المرقمة 40 . فقد سبق أن أوضحنا فيما مضى ان مثل هذا البرنامج سيستمر بطلب قراءة جديدة (عند استخدام وسيلة الادخال بواسطة جملة INPUT) الى ان نوقفه باستخدام مفتاح الايقاف STOP . اما هنا فقد توقف من حاله اضطرارياً عند نفاذ كافة القراءات التي يستطيع قراءتها والتي أعطيناه اياها بواسطة جملة البيانات المرقمة 50 . لذلك فقد وصل البرنامج اثناء تنفيذه الى الجملة 11 للمرة السادسة وحاول ان يجد درجة الحرارة المقابلة لتلك القراءة من خلال الجملة 50 فلم يجدها لانها قد نفذت (فقط خمس قراءات) ولهذا السبب توقف توقفاً اضطرارياً واعطى اشارة عن موقع التوقف وهو الجملة 11 وعن سبب التوقف وهو نفاذ البيانات .

وهذه وسيلة مناسبة لادخال البيانات التي نعرفها قبل البدء بالبرنامج . وهكذا تجد اننا توصلنا الى ثلاث وسائل لادخال البيانات . الاولى من خلال جملة الإحلال الحسابية ويجبذ استخدامها اذا كان هناك قراءة واحدة او قراءتين مثلاً . والثانية من خلال جملة ادخل فيما اذا كان لدينا بيانات كثيرة نريد ادخالها واحدة واحدة عند كل خطوة من خطوات تنفيذ البرنامج والوسيلة الثالثة هي من خلال جملة اقرأ مع جملة بيانات، ويجبذ استخدامها عند وجود بيانات ثابتة معروفة قبل البدء بتنفيذ البرنامج . دعنا نأخذ مثالاً آخر . ولكن هذه المرة نجبذ أن يكون لديك قليل من المعلومات في علم الاحصاء .. لا عليك إن لم يكن لديك اية معلومات فسنحاول مساعدتك .

لديك ستة قراءات تمثل درجات طلبة في صف ما مثلاً ، والمطلوب ايجاد متوسط هذه القراءات والانحراف المعياري لهذه الدرجات .

اظن انك تعرف ماهو المتوسط . ولكن ربما لاتعرف ، هو الانحراف المعياري ، انه ببساطة

$$\text{الانحراف المعياري لدرجات الطلبة} = \sqrt{\frac{(\text{الدرجة الاولى} - \text{متوسط الدرجات})^2 + (\text{الدرجة الثانية} - \text{متوسط الدرجات})^2 \ldots}{\text{عدد الدرجات}}}$$

لنفرض ان القراءات الست سميت A ، B ، C ، D ، E ، F على التوالي وان المتوسط هو V والانحراف المعياري هو S . وان درجات الطلبة كانت 95 ، 85 ، 40 ، 77 ، 60 ، 62
سيكون البرنامج كالاتي :

```
10 READ A, B, C, D, E, F
20 V = (A + B + C + D + E + F) /6
30 S = SQR (( (A-V) ^2 + (B - V) ^2 + (C - V) ^2
   + (D - V) ^2 + (E - V) ^2 + (F - V) ^2) / 6)

40 DATA 95, 85, 40, 77, 60, 62
50 PRINT "AVERAGE =", V, "STANDARD DEVIATION =", S
```

ماالجديد في هذا البرنامج ؟ انه استخدام SQR واظنك ستعرف ان لها علاقة بالجذر التربيعي لان المعادلة التي تحسب الانحراف المعياري تحتوي على الجذر التربيعي . بالطبع كان بالامكان تجاوز ذلك برفع المقدار للاس نصف (0.5) كما فعلنا في ايجاد مساحة المثلث الذي عرفت اضلاعه الثلاثة في الفقرة 14 . ولكن لم لاننتهز هذه الفرصة للتعرف على هذه الدالة الجديدة ذات الحروف الثلاثة والمأخوذة من كلمة Square Root والتي تعني الجذر التربيعي ؟ انها وسيلة لايجاد الجذر التربيعي لعدد (طبعاً يجب ان يكون موجباً) او لمقدار حسابي بعد ايجاد قيمته . ويتم ذلك ببساطة :

جذر 4 مثلاً SQR (4) وذلك بمجرد كتابة الدالة SQR ووضع العدد بين قوسين ، واذا كان العدد هو حاصل احتساب مقدار معقد كما في مثالنا اعلاه فإننا نضع ذلك المقدار بين القوسين .

اذاً الدالة (تدعى دالة مكتبية) هي عبارة عن علاقة معروفة مع المقدار المطلوب ايجاد دالته وتكون هذه العلاقة في اغلب الاحيان رياضية (هناك علاقات غير رياضية سنأتي على ذكرها في الاجزاء الاخرى من الكتاب ان شاء الله) وتكون ثابتة ومعروفة من قبل الحاسبة فهي مثل ايعاز بايجاد الجذر التربيعي او غير ذلك في الدوال التي سنأتي على ذكرها .

هذا مع العلم انك تستطيع ان تصنع لنفسك دالة او دوال خاصة لبرنامجك تدعى (دوال معرَّفة من قبل المستخدم) وسنأتي على ذكرها في المستقبل ان شاء الله .

والان دعنا نعود الى برنامج درجات الطلبة ومتوسطها والانحراف المعياري حيث نجد اننا في الجملة 10 قرأنا الدرجات بمساعدة جملة البيانات المرقمة 40 والتي كانت ست درجات وقام البرنامج باعطاء الدرجة الاولى 95 للمتغير A والثانية 85 للمتغير B والثالثة 40 للمتغير C وهكذا بحسب ورودها ضمن الجملة وبالتسلسل نفسه .

وفي الجملة 20 تم ايجاد متوسط القراءات وذلك بجمعها وقسمه حاصل الجمع على عددها وهو ستة .

اما الجملة 30 فقد احتسبت الانحراف المعياري بايجاد الفرق بين كل درجة والمتوسط الذي حسبناه V ثم تربيع ذلك (رفعه للاس 2) ثم جمع هذه المربعات وقسمة الناتج على عددها وهو 6 ثم ايجاد الجذر التربيعي لحاصل القسمة هذا بالضبط كما مطلوب من خلال القانون الذي ذكرناه في بدء المسألة . ثم تم طبع الناتجين من خلال الجملة 50 .

لاحظ ان جملة البيانات 40 قد وردت قبل جملة الطبع 50 . وبالحقيقة ان موقع جملة البيانات يمكن ان يكون في اي موقع من البرنامج حسب اختيارنا وليس هناك اية دلالة لموقعها سوى ان البرنامج يربط بينها وبين جملة اقرأ READ . ولكن يجب الحذر اذا كان في البرنامج اكثر من جملة قراءة او اكثر من جملة بيانات حيث يجب ان ترد بحسب ورودها اثناء تنفيذ البرنامج فمثلاً كان بالامكان ان يكتب البرنامج السابق كالآتي

```
5     READ A, B, C      } بدل جملة 10 سابقاً
15    READ D, E, F      }
20    V = ...           }
30    S = ...           } كما هي سابقاً
50    PRINT ...         }
55    DATA 95, 85       } بدل جملة 40 سابقاً
65    DATA 40, 77, 60, 62 }
```

لاحظ أننا استعضنا عن جملة 10 بجملتين : 5 و 15 وقسمت المتغيرات بينهما لكل واحدة ثلاثة متغيرات .

أما جملة البيانات 40 فقد استعيض عنها بالجملتين 55 و 65 وقسمت البيانات بينهما بحيث أخذت الاولى قيمتان والثانية أربع ، وبالطبع فإن هذا الاسلوب لامبرر له رغم أنه صحيح طالما أن تسلسل المتغيرات الستة هو تسلسل الدرجات الست بالضبط ، رغم تقسيم كل من المتغيرات والبيانات الى جملتين لكل منها . وقد بقيت الجمل 20 و 30 و 50 كما كانت .

والآن . . لنحاول تلخيص ما تعلمناه في الفقرات السابقة . لقد تعلمنا عدداً لا بأس به من الايعازات هي ايعاز الطبع PRINT وايعاز التنفيذ RUN وايعاز الاحلال افرض LET وايعاز الادخال INPUT وايعاز العرض والتسطير LIST وايعاز اقصد GOTO وايعاز القراءة READ وجملة البيانات DATA والدالة المكتبية للجذر التربيعي SQR .

كما تعلمنا وسائل برمجية متعددة منها طبع الشرح او الايضاح وتنفيذ وعرض كل او جزء من البرنامج والقيام بالعمليات الحسابية كالجمع والطرح والضرب والقسمة والرفع الى الاسس والتعويض في المقادير الحسابية المعقدة وعرفنا الفرق بين العدد الثابت والمتغير وعرفنا أن هناك متغيرات حرفية تستخدم معها علامة العملة $ وتعلمنا كيف نستخدم معداداً للعد وماهي انسب طريقة للتعامل مع البيانات القليلة والبيانات العديدة المتغيرة منها والثابتة . وتعلمنا استخدام بعض المفاتيح الخاصة مثل مفتاحي الدالتين الخامسة F_5 والرابعة F_4 ومفتاح الايقاف STOP . سنحاول الآن جمع شتات هذه المعلومات ببعض الأمثلة .

22 . أمثلة

ـ مثال (1) ـ اربع متوازيات مستطيلات كل منها عرفت اضلاعه . يطلب كتابة برنامج يقوم بحساب حجومها ومساحاتها السطحية وطبع ذلك بشكل جدول .
الحل : اذا فرضنا أن اطوال اضلاع متوازي المستطيلات هي A و B و C فإن حجمه $\overline{V}$ هو حاصل ضرب الاضلاع مع بعضها أي v = abc
واذا كانت مساحته السطحية هي S فانها تساوي

S = 2 (ab + bc + ca)

وهكذا وباستخدام جملة القراءة والبيانات وتحويل التحكم : اقصد يمكننا كتابة البرنامج على النحو الآتي :

```
10      READ A و B و C
20      V = A * B * C
```

```
30   S = 2 * (A * B + B * C + C * A)
40   PRINT "SIDES ARE:" ,A, B, C ,"VOLUME = "
                , V , "SURFACE AREA =",S

50   G OT O    10
60   DATA      3,6,2
61   DATA      4,3,5
62   DATA      7,2,3
```

لاحظ اننا وضعنا ثلاث جمل للبيانات وذلك لغرض فصل كل متوازي مستطيلات عن الاخر بحيث تعطى اضلاعه بجملة مستقلة . كما اننا حاولنا أن نضع ارقام لهذه الجمل بشكل متسلسل 62,61,60 لتجنب الخطأ ربما بسبب ادخال جملة جديدة فيما بينها تحوي بيانات اخرى فيلتبس الموضوع على الحاسبة . ويرتكب بعض المبرمجين مثل هذا الخطأ اذا ما كتبوا برنامج منذ فترة طويلة ثم قاموا بتعديله بعد ذلك . وهكذا فإن على من يكتب برنامجاً أن يتقن تلك الكتابة لتجنب اخطاء المستقبل ... أن الله يحب اذا عمل احدكم عملاً أن يتقنه .. وسيمر بنا فيما بعد اسلوب كتابة ملاحظات داخل البرنامج للاستفادة منها في المستقبل .

واخيراً بعد تنفيذ البرنامج اعلاه سوف تحصل على النواتج الآتية :

```
SIDES ARE :3   6   2 VOLUME = 36 SURFACE AREA  = 72
SIDES ARE :4   3   5 VOLUME = 60 SURFACE AREA  = 94
SIDES ARE :7   2   3 VOLUME = 42 SURFACE AREA  = 82
```

ـ مثال (2) ـ كان يوم 1/ 1/ 1986 يوم الاربعاء . يطلب عمل تقويم لشهر كانون الثاني 1986 يبين اسم اليوم والتأريخ .

لنفرض أن تأريخ اول يوم من الشهر هو I . للاسبوع الاول يمكننا كتابة البرنامج على النحو الآتي :

```
10    I = 1
20    PRINT     "WEDENSDAY",I
30    PRINT     "THURSDAY",I + 1
40    PRINT     "FRIDAY", I + 2
50    PRINT     "SATURDAY", I + 3
```

```
60      PRINT       "SUNDAY" , I + 4
70      PRINT       "MONDAY" , I + 5
80      PRINT       "TUESDAY" , I + 6
```

وهكذا سوف تطبع تواريخ الايام السبعة الاولى من الشهر . ولكن ماذا عن باقي الايام ؟
لماذا لانحاول ان نجمع للمتغير I سبعة لكي نحصل على الاسبوع الثاني ونعيد الكَّرة ؟ مثلاً

```
90      I = I + 7
100     GO TO 20
```

جرِّب هذا البرنامج . ستلاحظ ان ايام الاسبوع الثاني قد ظهرت ايضاً وليس ذلك فحسب ، بل ظهرت ايام الاسبوع الثالث والرابع لان هاتين الجملتين تصلحان لكافة هذه الأسابيع .

ولكن ماذا يحدث بعد ذلك ؟ سترى في الاسبوع الخامس ان اليوم الاول : الاربعاء تأريخه 29 والخميس 30 والجمعة 31 والسبت 32 والاحد 33 والاثنين 34 والثلاثاء 35 ، وهكذا يستمر الاسبوع السادس ... الخ ولن يتوقف البرنامج الا اذا ضغطت على مفتاح الايقاف STOP .

ولكن الا يمكن ان يتوقف البرنامج ذاتياً حينما يصل نهاية الشهر حيث نعلم ان كانون الثاني يحوي 31 يوماً فقط ؟

بالطبع يمكن ذلك ، وهناك وسائل بعضها بسيط وسيمر بنا في المستقبل . ولكن في ضوء ما أخذنا من معلومات لحد الآن الا يمكن ان نحتال على البرنامج ونوقفه بصورة تلقائية حينما ينتهي الشهر ؟ فكر في الموضوع قليلاً قبل ان تستمر في قراءة السطور الآتية ... وحاول استعمال قابلياتك العقلية .. فنعمة العقل أفضل النعم التي انعم الله بها على الانسان .. وعليك استخدامها افضل ما تستطيع فذلك هو شكر تلك النعمة .. ولا تستعجل في الاستمرار في القراءة قبل ان تفكر في المسألة .. فالعجلة من الشيطان .. واذا لم تهتد الى الجواب فاقترح عليك مراجعة جملة القراءة READ والبيانات DATA وكيف اوقف برنامج حساب درجات الحرارة المئوية بعد خمس قراءات فقط لنفاذ البيانات .
هل اهتديت الى وسيلة ؟

لنحاول معاً خطوة .. خطوة .. ماذا نريد ؟ نريد ايقاف البرنامج بعد ان ينفذ الجملة 40 للمرة الخامسة حيث سيعطي في المرة الاولى : الجمعة 4 وفي الثانية :

الجمعة 11 وفي الثالثة : الجمعة 18 وفي الرابعة : الجمعة 25 وفي الخامسة : الجمعة 31. ولانريد ان يصل يوم السبت الذي يليه لاننا لانريد ان نرى التأريخ الخاطيء 32.

ماذا يحدث لو وضعنا جملة جديدة بين 40 و 50 مثلاً برقم 45 تقوم بقراءة شيء ما؟ ولنفرض انه رقم الاسبوع الذي سنسميه W ونعطي فقط ارقام لاربعة اسابيع لاغيرها . وعند الانتهاء من هذه الاسابيع الاربعة التي تنتهي يوم الجمعة سوف يتوقف الحل لنفاذ البيانات كما سبق وان فعلنا في بيانات درجات الحرارة . أي :

```
45      READ    W
110     DATA    1, 2, 3, 4
```

جرب البرنامج الآن . . اولاً حاول تسطير البرنامج او عرضه بواسطة F_4 سترى ان هاتين الجملتين دخلتا ضمن البرنامج . نفذ البرنامج بواسطة F_5 سترى ان ايام الشهر كلها ستظهر واحدة بعد الاخرى بشكل اسم اليوم وتأريخه وتتوقف بعد اليوم الحادي والثلاثين (يوم الجمعة) معلنة انتهاء البيانات

```
OUT OF DATA    IN    45
```

تمرين

والآن . . هل تستطيع اعادة كتابة البرنامج لكي يصلح لشهر شباط 1986 الذي يبتديء يوم السبت وعدد ايامه 28 يوماً؟ حاول ذلك . لاحظ اننا لم نستفد من قيم رقم الاسبوع W وكأن البرنامج سينفذ كما هو مهما كانت الارقام المدرجة في جملة البيانات 110 طالما كان عددها اربعة فقط .

(ملاحظة عن الحل موجودة في نهاية الكتاب) .

ـ مثال (3) ـ دوري لكرة القدم ، يشارك فيه عدد من الفرق الرياضية . لنفرض ان عددها 6 . كل فريق يجب أن يدخل مباراة مع كافة الفرق الاخرى ، كم مباراة ستجري؟ عدد المباريات هو

$$1 + 2 + 3 + 4 + 5 = 15 \text{ مباراة}$$

المطلوب : ادخال اسماء الفرق الستة . ثم عمل جدول بالفرق التي تتبارى بحيث تكون هناك مباراة واحدة باليوم عدا يوم الجمعة حيث لاتجري اية مباراة . افرض ان أول يوم من ايام الشهر كان يوم السبت .

الحل : لنحاول اولاً قراءة اسماء الفرق فريقاً فريقاً . ولنفرض أن اسماء المتغيرات الحرفية لهذه الفرق الستة هي على التوالي A$ و B$ و C$ و D$ و E$ و F$.

```
10      INPUT "TEAM1", A$
11      INPUT "TEAM2", B$
12      INPUT "TEAM3", C$
13      INPUT "TEAM4", D$
14      INPUT "TEAM5", E$
15      INPUT "TEAM6", F$
```

ماذا عن مواعيد المباريات؟ ومن يتبارى مع من؟

لنفرض أن المبارات في اليوم الاول تجري بين الفريق رقم 1 ورقم 2 وفي اليوم الثاني بين الفريقين 3 و 4 .... وهكذا يمكننا ترتيب جدول بذلك ولكن كيف ندخل ذلك الجدول الى الحاسبة؟ الا يمكن اجراء ذلك بواسطة جملة القراءة READ والبيانات DATA؟
لنحاول ذلك

```
20      READ I, N, M
30      PRINT "DATA:", I, "TEAMS NUMBERS", N,
    "AND", M
40      G Θ T Θ 20
50      DATA 1,1,2, 2,3,4, 3,5,6, 4,2,3, 5,4,5,
          6,6,1, 8,1,3 9,2,4, 10,3,5, 11,6,4,
          12,1,4, 13,2,5 15,3,6, 16,1,5, 17,2,6
```

وهكذا ترى أن كل ثلاث قراءات تكون مجموعة وقد وضعت الفراغات بينها للايضاح : القراءة الاولى هي التاريخ والثانية والثالثة هما رقما الفريقين المتباريين .

وبالطبع فإن هذه الارقام لم يتم الوصول اليها وادخالها في المسألة الابعد رسم مربع يشبه جدول الضرب لاسقاط الفرق المتبارية عمودياً وافقياً مع تجنب أن تلعب الفرقة مع نفسها أو أن تلعب مع غيرها اكثر من مرة واحدة . ولكن كيف نعرف اسماء هذه الفرق؟

يمكن اضافة جملة مثلاً برقم 15 لادراج اسماء الفرق وارقامها مثلاً :

```
15 PRINT 1, A$, 2, B$, 3, C$, 4, D$, 5, E$, 6, F$
```

وهكذا سوف تطبع ارقام الفرق واسماؤها بواسطة هذه الجملة .

لنحاول اولاً عرض البرنامج بواسطة F4 ثم تنفيذه بواسطة F5 سترى أن البرنامج سيطالبك باسم الفريق الاول والذي يمكنك ادخال اسمه مثلاً فريق الجيش ARMY ثم الفريق الثاني فالثالث .. الى السادس . وبعد ذلك سيقوم البرنامج بطبع قائمة بارقام الفرق واسمائها كما ادخلتها ثم بعد ذلك يدرج لك تواريخ الايام اعتباراً من اليوم الاول ويعقبه رقما الفريقين المتباريين على النحو التالي :

```
DATE  :1    TEAMS NUMBERS  1    AND    2
DATE  :2    TEAMS NUMBERS  3    AND    4

DATE  :8    TEAMS NUMBERS  1    AND    3
DATE  :9    TEAMS NUMBERS  2    AND    4

DATE  :17   TEAMS NUMBERS  2    AND    6
OUT   OF    DATA    IN     20
```

وهكذا انتهت البيانات فتوقف البرنامج واعطى اشارة بذلك . لابد وانك ستلاحظ اننا نحن الذين قمنا بتزويد هذا البرنامج بمواعيد المباريات (التاريخ) وارقام الفرق المتبارية . ولم يقم البرنامج سوى بطبع ذلك على شكل جدول وكذلك اسماء الفرق . ولكن لا تنسى انك لا تزال مبتدئاً في البرمجة ورغم أن الحاسبة الالكترونية بامكانها وضع جدول للمباريات وفق برنامج يعد لذلك خصيصاً ، الا أن مثل ذلك البرنامج لا يزال معقداً بالنسبة لمرحلتنا هذه ويحتاج الى معرفة المزيد من الايعازات والخبرة في اساليب البرمجة . سنحاول في المستقبل أن شاء الله الرجوع الى هذا المثال لا يضاح المزيد من الاساليب البرمجية التي يمكننا بواسطتها الحصول من الحاسبة الالكترونية على مساعدات اكثر وحل لمسائل اعقد .

23 . تمارين

1. اكتب برنامجاً يقرأ عدد سكان قطر ما في سنة ما ونسبة الزيادة السنوية بالسكان ويقوم بحساب عدد السكان المتوقع خلال السنوات العشرة التالية .
2. اذا علمت أن شخص قد تقاضى راتبه على النحو الآتي :

   A قطعة نقود من فئة 10 دينار .
   B قطعة نقود من فئة 1 دينار .

C قطعة نقود من فئة 50 فلساً .
احسب راتب الشخص وفق برنامج يقرأ مقادير C, B, A ويعطي الراتب لخمسة اشخاص .

3. جسم يسير بتعجيل عرفت سرعته الابتدائية وتعجيله . المطلوب حساب سرعته بمضي الزمن . اكتب برنامجاً يقرأ السرعة الابتدائية والتعجيل ويعطي السرعة كل خمس ثواني لمدة 20 ثانية .

24 . الانتقال الشرطي : اذا .. اذن IF...THEN

في كافة البرامج والامثلة التي مرت بنا لحد الآن كانت البرامج اثناء تنفيذها تنفذ جملة جملة حسب تسلسل ارقامها الا اذا وصلت جملة اقصد GOTO فعند ذلك ينتقل التنفيذ الى الرقم الذي تشير اليه تلك الجملة ويستمر التنفيذ بعدها الى الجملة التي تليها رقماً ، وهكذا ... أن للغة البرمجة بيسيك كما لغيرها من لغات البرمجة الاخرى القابلية على الانتقال الى اية جملة مقصودة عند تحقق شرط معين وفيما عدا ذلك يستمر التنفيذ كما هو مقرر . تدعى جملة الانتقال الشرطي اذا .. اذن IF...THEN وسندعوها للسهولة جملة اذا .

كما يظهر من تسمية الجملة أنها شرطية لذا تحتاج الى شرط بين شقيها (بين IF,THEN) وهذا الشرط يكون بشكل رياضي اما أن يتحقق (يكون صحيحاً) او لا يتحقق (يكون غير صحيح) . فاذا ما تحقق نفذت جملة اذا جزءها الذي يعقب THEN وهو الانتقال الى جملة محددة الرقم . اما اذا لم يتحقق الشرط فإن التنفيذ ينتقل الى الجملة التي تلي جملة اذا في تسلسلها .

لنفرض أن المطلوب هو الانتقال الى الجملة المرقمة 70 في حالة تحقق شرط معين ، وليكن حينما تكون B اكبر من 5 . حين ذلك تكون صيغة الجملة :

```
20    IF B > 5    THEN    70
```

لاحظ اننا استخدمنا علامة الاكبر بين 5,B ويكون B>5 هو الشرط الذي يمكن أن يأخذ أحدى القيمتين : اما صحيح (اذا كانت B اكبر من 5) أو خطأ (اذا كانت B أصغر أو تساوي 5) .

وحيث أن هذه اول مرة نصادف فيها جملة شرطية لذلك علينا أن نتعرف على العلاقات التي يمكن أن تحتويها تلك الجمل . أن صيغة الشرط في جملة اذا هي :

[مقدار] [علاقة] [مقدار]

المقدار يمكن أن يكون عدداً أو متغيراً او مقداراً حسابياً معقداً اما العلاقة فيجب أن تكون احدى العلامات الآتية :

يساوي	=
اكبر	>
اكبر أو يساوي	=>
أصغر	<
أصغر أو يساوي	=<
أكبر أو أصغر (لا يساوي)	><

لنأخذ بعض الإمثلة على ذلك :

```
2    IF    X = 10    THEN    200
```

وتعني اذا كانت قيمة X تساوي 10 ينتقل التحكم الى الجملة 200 . وفيما عدا ذلك (اذا كانت X اكبر من 10 أو أصغر منها) ينتقل التحكم الى الجملة التالية لهذه الجملة .

```
30    IF    X < 20    THEN    80
```

اذا كانت X أصغر من 20 ينتقل التحكم الى الجملة 80 ، اما اذا كانت X تساوي 20 أو اكبر من ذلك فيتم الانتقال الى الجملة التي تلي الجملة 30 .

```
40    IF    R => B ^ 2    THEN    100
```

اذا كانت قيمة R أكبر من مربع B او مساوية لها فإن التحكم ينتقل الى الجملة 100 . اما فيما عدا ذلك (اذا كانت R أصغر من مربع B) فيستمر التحكم الى الجملة التالية . لاحظ أن مثل هذه الجملة يمكن أن توضع باشكال متعددة مثلاً :

```
41    IF    R-B ^ 2 => 0    THEN    100
42    IF    B ^ 2 < R       THEN    100
```

فالجملة 41 هي نفسها الجملة 40 .. اذا ما طرحنا من طرفي المتباينة (أو المتراجحة كما تدعى احياناً) المقدار B ^ 2 حيث تبقى قيمتها ثابتة . وفي الجملة 42 تم عكس تسلسل ورود طرفي المتراجحة مما استوجب أن تعكس العلاقة (من – أكبر او يساوي – الى اصغر) .

وهكذا كانت الجمل 42,41,40 هي الجمل نفسها دون فرق .

```
50    IF    A >< 0    THEN    40
```

اذا لم تساوي قيمة A الصفر (اي كانت اكبر من الصفر او اصغر منه) ينتقل التحكم الى الجملة 40 . اما اذا كانت قيمة A تساوي صفراً فإن التحكم ينتقل الى الجملة التالية .

```
60      IF      N$ = "AHMED" THEN      200
```

اذا كان المتغير الحرفي N$ يساوي الاسم « احمد » فإن التحكم ينتقل الى الجملة 200 . وفيما عدا ذلك يستمر نحو الجملة التالية . لاحظ أن المتغير الحرفي N$ يجب أن يكون قد أخذ قيمة (او اسماً معيناً) وفي هذه الجملة يراد التأكد من ان الاسم ذلك هو « احمد » ام لا . وهذه وسيلة فعالة لكي تتعرف الحاسبة على الاسماء . مثلاً يمكن أن تكتب برنامجاً لا تريد أن يستخدمه غيرك وحينما يبتدىء البرنامج يسأل المستخدم عن اسمه فإن كان الجواب غير الاسم المحدد في البرنامج انتقل التحكم الى جملة خاصة مثلاً جملة ايقاف STOP . اما اذا كان الاسم هو الاسم الصحيح المحدد في البرنامج فان الحل يستمر كما هو مقرر في البرنامج .

كما تجدر الاشارة الى أن أي خطأ في كتابة الاسم سوف يؤدي الى اعتبار الاسم مختلف : مثلاً اذا كتبت AHMAD بدل AHMED فإن الجملة 60 سوف تعتبر الاسم مختلف ولا تستجيب بالانتقال الى الجملة 200 المقصودة .

وهكذا تعلّمنا الحاسبة الالكترونية على الدقة ولن يتم ذلك الا بالتقيد بانظمة واضحة وبسيطة . وتجدر الاشارة الى أن كثيراً ما تؤدي كتابة البرامج او البيانات بخط رديء الى خلط بين الحروف والارقام وخاصة القريبة من بعضها مثلاً حرف O مع الصفر وحرف z مع الرقم 2 وغيرها .

الآن . بعد أن تعرّفنا على جملة (اذا) هذه يمكننا استخدامها في كثير من الامثلة التي مرّت بنا وكنا نلاقي صعوبة في ايقاف البرنامج عند حد معين حيث كنا نحتال لذلك بانهاء البيانات ومن ثم يضطر البرنامج للتوقف . أما الآن فبامكاننا ايقاف البرنامج متى نشاء باستخدام جملة (اذا) .

لنرجع الى مثال تحويل خمس قراءات لدرجة الحرارة بالقياس الفهرنهايتي الى القياس المئوي . يمكن الآن أن نضع البرنامج كالآتي :

```
5       I= 1
11      READ      F
20      C = (F-32) * 5 / 9
30      PRINT     I,C
35      I = I + 1
41      IF      I,≤6 THEN 11
50      DATA      14,23,32,41,50
```

لاحظ أننا لم نفعل شيئاً سوى استبدال الجملة 41 بدل الجملة 40 السابقة في البرنامج الذي مر في الفقرة 21 والتي كانت جملة أقصد GO TO .
الآن الجملة 41 تقوم بالتأكد من قيمة المعداد I هل هي اصغر من ستة ام لا ؟ فاذا كانت اصغر من ستة (من واحد الى خمسة) ينتقل التحكم في كل مرة الى الجملة 11 لغرض استلام قراءة جديدة (من البيانات) . أما اذا بلغت قيمة I ستة فان الشرط في الجملة 41 اصبح خاطئاً ومن ثم ينتقل الحل الى الجملة التالية اي الى نهاية البرنامج . وسترى عند تنفيذ هذا البرنامج إنه يحول الدرجات الفهرنهايتية الخمس الى مئوية ثم يتوقف دون أية اشارة بوجود نقص في البيانات .

لاحظ ان الشرح الآنف الذكر يصلح لمثالين آخرين سبق شرحهما احدهما هو مثال كتابة ايام الاسبوع لشهر كانون الثاني 1986 والذي حاولنا ايقافه في نهاية الشهر باضافة الجملتين 45 و 110 . الا يمكن حذفهما ؟ واضافة بدلاً منهما الجملتين ؟ :

```
47      IF I + 2 = 31 THEN 105
105     STOP
```

والآن . عليك أن تحاول تحوير برنامج مثال كرة القدم السابق (مثال 3-الفقرة 23) لكي يتوقف بصورة تلقائية عند انتهاء البيانات دون اضطراب .

25. مثال على جملة اذا

سنحاول كتابة برنامج لاعلان نتائج الطلبة (40 طالباً) ، ولنفرض أن عدد الدروس كان عشرة . وإننا نريد أن ندخل اسم الطالب ودرجاته لعشرة دروس ثم يقوم البرنامج بكتابة نتيجة الطالب وهي احد ثلاث حالات : أ) ناجح .. اذا كانت درجاته كلها 50 فما فوق .
ب) مكمل .. اذا كانت درجة درس واحد او درسين فقط اقل من خمسين .
جـ) راسب .. اذا كان هناك ثلاث دروس فأكثر اقل من خمسين .

الحل : سنفرض ان اسم الطالب يدخل في المتغير الحرفي N$ وان درجاته تدخل واحدة واحدة حتى الدرجة العاشرة وسنحجز المعداد I لكي يحسب تسلسل هذه الدرجات .

ثم نحجز المتغير I لعدد الدروس التي يحرز الطالب فيها درجة أقل من 50 وبعد ادخال درجات الطالب كافة ندقق عدد الدروس هذه . فإن بلغ 3 فأكثر اعتبر راسباً وان كان 1 أو 2 اعتبر مكملاً ، اما اذا كان العدد يساوي صفراً فيعتبر ناجحاً .

وهكذا ننتقل الى الطالب الثاني فالثالث ، لذا نحتاج الى معداد لعدد الطلبة لنفرض إنه المتغير K : فاذا بلغ عدد الطلبة اربعون طالباً توقف الحل .

هل تريد بعد ذلك احصائيات عن عدد الناجحين والمكملين والراسبين في الصف كله؟ اذا اردت ذلك احجز ثلاث متغيرات ، مثلاً P للناجحين و M للمكملين و F للراسبين وسترى كيف نحسب ذلك .
لنبدأ متكلين على الله :

نقوم اولاً باعطاء قيم ابتدائية لكافة المتغيرات :

```
100     K = 1
110     I = 1
120     J = O
```

لماذا؟ نبدأ بالطالب الاول رقمه K = 1 وبالدرجة الاولى للطالب I = 1 اما قيمة J وهي عدد الدروس التي رسب فيها فنبدأ بالصفر ، حيث نعتبره ناجحاً الا اذا ثبت رسوبه في احد الدروس كالمتهم الذي يعتبر بريئاً مالم تثبت ادانته ونحن لانعلم هل رسب في اي درس ام لا قبل قراءة درجاته. ثم ندخل اسم الطالب

```
130     INPUT N$
```

ودرجته الاولى لنفرض انها X

```
140     INPUT X
```

هل ان X تساوي 50 فأكثر ام لا ؟

```
150     IF X >= 50 THEN 160
```

اذا كانت X اصغر من 50 انتقلنا الى الجملة التالية الخاصة بالدروس الراسبة

```
155     J = J + 1
```

أما اذا لم تكن الدرجة اصغر من 50 (اي كانت 50 فما فوق) فينتقل الحل الى الجملة المرقمة 160 والتي سنحاول من خلالها الذهاب الى الدرجة التالية . ولكن قبل ذلك ينبغي لنا أن نتأكد هل إننا وصلنا الدرجة العاشرة .

وكيف لنا أن نعرف ذلك؟ يجب أن نضيف واحداً الى رقم الدرجة وهكذا فإن

```
160     I = I + 1
170     IF I <= 11 THEN 140
```

وهكذا فإننا انتهينا في هذه المرحلة من درجات الطالب العشر ، ولكن يجب أن نراعي درجات الطلبة التي تقل عن 50 والتي اشرنا اليها بالجملة 150 حيث تقصد الجملة 155 وهكذا فإن الجملة 155 ستكون بعد تنفيذها (زيادة عدد

دروس الرسوب بواحد) ينتقل الحل ايضاً الى الحملة 160 التي كان سيقصدها في حالة كون الدرس درس نجاح (الدرجة تساوي خمسين او اكثر) لذلك ما أن أضفنا واحداً الى عدد الدرجات التي تقل عن 50 وفق الجملة 155 حتى تنتقل الى الجملة 160 للانتقال الى الدرجة التالية من خلال الجملة 170 .

والآن .. بعد المرور على الجملة 170 للمرة العاشرة (بعدد الدروس) نكون قد انتهينا من درجات الطالب كافة ونريد أن نفحص وضعه هل هو ناجح أم مكمل أم راسب . وهكذا نبدأ بالرسوب : اذا كان J أكبر من 2 .

```
180     IF J > 2 THEN 250
```

والتي يمكننا أن نطبع فيها نتيجة الطالب الراسب

```
250     PRINT N$, "FAILED IN", J , "SUBJECTS"
```

والتي بعدها يجب أن ننتقل الى الطالب التالي :

```
260     K = K + 1
```

ثم ندقق هل لايزال هناك طلبة لم تقرأ درجاتهم؟

```
270     IF K < 41 THEN 110
```

اما اذا كان عدد الدروس التي قد رسب بها الطالب أقل من ثلاثة فإن الشرط من الجملة 180 سيكون غير صحيح لذلك ينتقل التحكم الى الجملة التالية 190 والخاصة بالطلبة المكملين .

```
190     IF J > 0 THEN 220
```

والتي يمكننا ان نطبع فيها نتيجة الطالب المكمل :

```
220     PRINT N$, "MAKE - UP IN", J, SUBJECT"
```

ثم بعدها ننتقل الى الطالب التالي من خلال

```
230     GO TO 260
```

كما سبق في حالة الرسوب .

اما الطالب الناجح فإن نتيجته يمكن أن تظهر بشكل

```
200     PRINT N$, "PASS"
```

ثم يتم الانتقال الى الطالب التالي

```
210     GO TO 260
```

وهكذا في نهاية الجملة 270 نكون قد انتهينا من البرنامج ، لذلك نضيف جملة لايقاف البرنامج

```
280     STOP
```

الآن بقي لنا أن نضيف أن شئنا حساب مجموع عدد الناجحين والمكملين والراسبين من خلال المتغيرات F, M , P على التعاقب .

نحاول اولاً تصفير هذه المتغيرات في بداية البرنامج مثلاً

```
70      P = O
80      M = O
90      F = O
```

ثم نحاول أن نجمع لهذه العدادات واحداً كلما مر التنفيذ على احدها مثلاً بعد الجمل 200 و 220 و 250 على التعاقب :

```
205     P = P + 1
225     M = M + 1
255     F = F + 1
```

وبعد الانتهاء من التنفيذ وقبل الجملة الاخيرة نكتب الناتج على النحو التالي :

```
275 PRINT "NO.OF PASSING", P, "NO. OF MAKE UP =",
M, "NO. OF FAILING =", F, "TOTAL NO. ="P + M + F
```

وهكذا ولغرض التأكد من صحة كافة العمليات ادرجنا في هذه الجملة عدد الطلبة الناجحين وعدد الطلبة المكملين وعدد الطلبة الراسبين ، ومجموع عدد الطلبة الذي يجب أن يكون 40 . ولو كان غير ذلك فيمكن أن يصلح البرنامج بمجرد تغيير الجملة 270 والتي يمكن حتى ان تكون عامة لاي عدد اذا أدخل عدد الطلبة في بدء البرنامج مثلاً

```
60 INPUT T
```

حيث T هو عدد الطلبة ومن ثم يجب أن تعدل الجملة 270 الى :

```
271     IF      K < T + 1 THEN      110
```

الان .. وقد انتهينا من حل المسألة ، نود أن نعترف بانها مسألة معقدة لهذه المرحلة ، وتحتاج الى المزيد من الايضاح .

لذلك نقترح ما يأتي : حاول أن تدخل المسألة على حاسبتك ثم تعرض البرنامج كاملاً بواسطة F_4 ثم تتفحصه جيداً وتفترض درجات وهمية لبضع طلبة بينهم الناجح والمكمل والراسب وتغذي تلك الدرجات للبرنامج لتتأكد من صلاحيته ، هذا ما نقترحه لك .

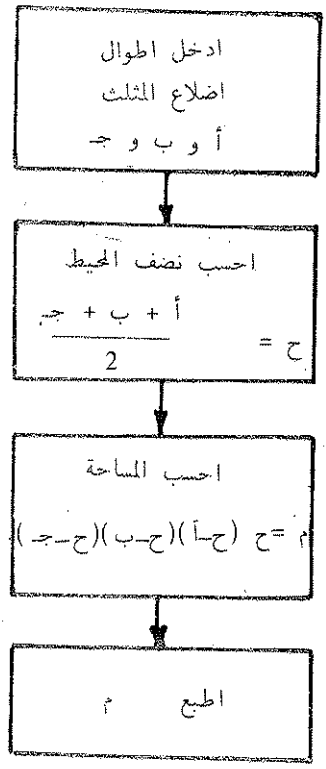
اما ما سنفعله الان فهو المزيد من الايضاح لخطوات الحل التي سرنا وفقها في كتابة البرنامج ولكن باسلوب ثانٍ بمساعدة ما يسمى بالمخططات الانسيابية .

26 . ما هي المخططات الانسيابية ؟

المخطط الانسيابي عبارة عن رموز مرسومة يربط بين رمز واخر سهم يدل على سريانه أو انسيابية المخطط ، له بداية وله نهاية . وبين البداية والنهاية سلسلة من هذه الرموز المرسومة تبين بشكل عام وبلا تفاصيل دقيقة وبوضوح تام طريقة حل المسألة .

لنحاول ايضاح ذلك بالرجوع الى مثال 2 في الفقرة 14 والذي يتعلق باحتساب مساحة مثلث عرفت اطوال اضلاعه الثلاثة .

المخطط الانسيابي للحل مبين في الشكل (5) أن خطوات الحل الاربعة تتكون من خطوة ادخال وخطوتي حساب وخطوة اخراج (طبع) هي كل ما يحتويه المخطط ، وبالطبع بضمنه القوانين أو المعادلات المستخدمة فيه .



شكل (5)

كل خطوة من الخطوات وضعت داخل مستطيل . هذا هو ابسط انواع المخططات الانسيابية .

ولكن .. هل يمكن أن يحتوي المخطط الانسيابي على اشكال اعقد من ذلك؟

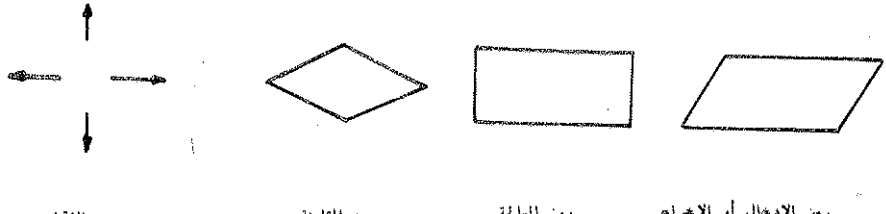
لقد استخدمنا في المثال املاه المستطيل كرمز عام لخطوات المخطط الانسيابي وعلى الرغم من أن محتويات المستطيل تكفي للدلالة على المقصود ، الا أن الكتب اصطلحت على استعمال رموز اخرى الى جانب المستطيل كي يبدو المقصود من العملية بمجرد رؤية الرمز المستعمل فيعرف منه فيما اذا كان المقصود عملية ادخال أو اخراج للمعلومات أو عملية مقارنة أو عملية معالجة حسابية . وسنقتصر هنا على اهم هذه الرموز ، وهي :

1- رمز ادخال المعلومات أو اخراج المعلومات : ويرمز لعمليتي الادخال والاخراج بمتوازي اضلاع يكتب في داخله ادخال اذا كانت العملية عملية ادخال للمعلومات أو اخراج اذا كانت عملية إخراج للمعلومات . ويتكرر هذا الرمز داخل المخطط حسب الحاجة الى عمليات الادخال والاخراج .

2- رمز المعالجة : يستعمل المستطيل للدلالة على عملية المعالجة الحسابية أو المنطقية أو الحرفية وقد تحتوي هذه المعالجة على مقادير بسيطة أو معقدة وقد يتكرر هذا الرمز عدة مرات حسب الحاجة الى استعماله .

3- رمز المقارنة : وهو على شكل معين يكون له ادخال واحد من الاعلى وعدد غير محدد من الاخراجات بحسب نوعية المقارنة . فهناك المقارنة ذات الاتجاهين والتي يجاب عنها بنعم أو بلا وهناك المقارنة التي تنتهي بثلاثة اتجاهات للاخراج مثلاً سالب أو صفر أو موجب وهناك المقارنة التي تنتهي بعدة اتجاهات للاخراج وتكون محتويات المعين المقادير أو الكميات أو المتغيرات المطلوب مقارنتها .

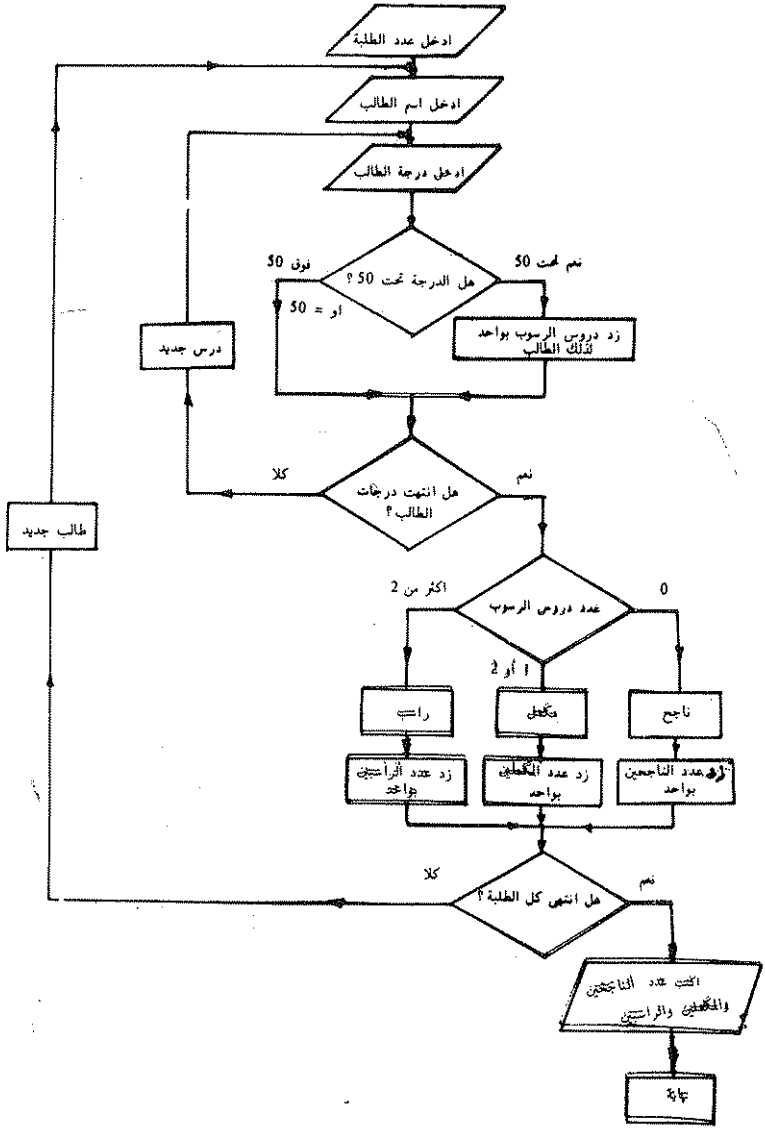
4- رمز الاتجاه : ويستعمل السهم للربط بين رموز المخطط الانسيابي . وتبرز اهمية رموز الاتجاهات في المخططات المعقدة والمتشابكة .



شكل (6)

27 امثلة على المخططات الانسيابية :

والان لنرجع الى مسألة درجات الطلبة ونحاول أن نرسم لها مخططاً انسيابياً . ولغرض الايضاح سنبدأ بمخطط مبسط كما في الشكل (7) ثم ننتقل الى المخطط النهائي في ضوء خبرتنا في البرمجة :



الشكل (7)

الادخال والاخراج يحدد بواسطة متوازي اضلاع ، وهو واضح في الشكل في ادخال عدد الطلبة وفي ادخال اسم الطالب وفي ادخال درجاته واحدة واحدة وفي اخراج نتيجته ناجح أو راسب أو مكمل وفي اخراج الاحصائية النهائية لعدد الناجحين والمكملين والراسبين .

أما رموز المقارنة فتلاحظ في اربعة مواضع : للتحقق من أن درجة الطالب هي تحت 50 أم لا وفي التأكد من الانتهاء من كافة الدروس للطالب الواحد وفي التأكد من الانتهاء من كافة الطلبة وفي تحديد عدد الدروس التي رسب بها الطالب ومن ثم تحديد نتيجته وفقها .

اما رموز المعالجة فتجدها هنا وهناك في زيادة عدد دروس الرسوب أو الانتقال الى درس جديد أو طالب جديد أو في زيادة عدد الناجحين أو الراسبين أو المكملين .

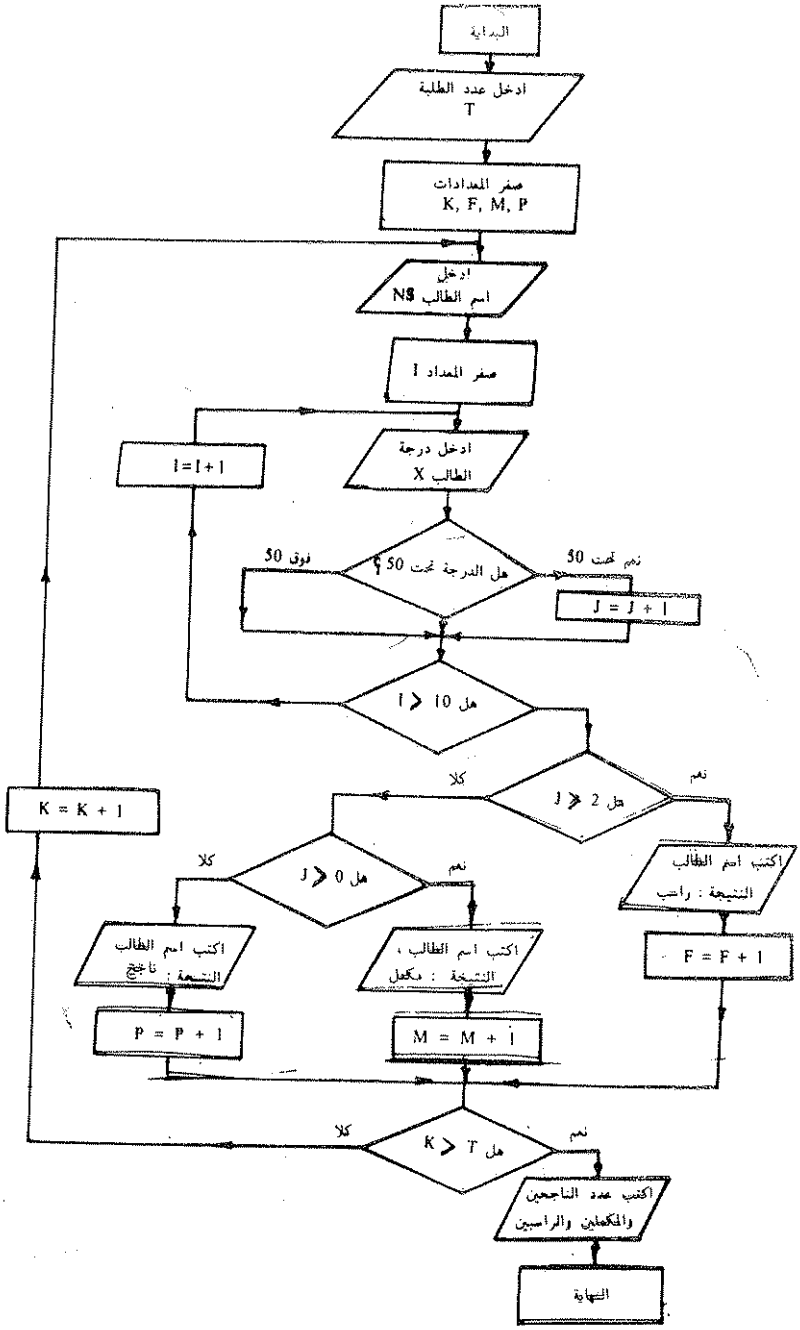
ويلاحظ من المخطط أن هناك حلقتان احداهما للانتهاء من درجات الطالب درجة درجة والثانية للانتهاء من كافة الطلبة طالباً طالباً . وكلا هاتين الحلقتين تنتهيان برمز مقارنة : اما أن يعيدها الى بدايتها أو أن يخرج الحل منها .

ويلاحظ من المخطط ايضاً اننا عند الاشارة في رموز المعالجة الى زيادة عدد دروس الرسوب أو الانتقال الى درس جديد أو طالب جديد أو غير ذلك لم نحدد كيف سيتم ذلك؟ ولا المتغيرات المستخدمة لهذه الاغراض .

كما يلاحظ أن رمز المقارنة لتحديد الاتجاه وفق عدد دروس الرسوب ذي ثلاث اتجاهات . ويجب أن يتم ذلك برمجياً باسلوب خاص حيث كما علمنا أن جملة اذا التي درسناها ذات اتجاهين : احدهما عند تحقق المقارنة بشكل صحيح والاخر عند عدم تحققها .

لذلك فإن المخطط الانسيابي التام الذي يقرب الحل من كتابة البرنامج بشكل دقيق يجب أن يحتوي على حل لهذه التساؤلات قدر الامكان .

والحق يقال أن المخطط الانسيابي يجب أن يوضع بشكل دقيق قبل البدء بكتابة البرنامج . الا أن اكثر المبرمجين يجدون من الافضل رسم مخطط انسيابي عام (غير دقيق) اولاً ثم البدء بالبرنامج وبعد ذلك يمكن وضع المخطط الانسيابي النهائي والذي يمكن أن يساعد في تصحيح او تحسين جزء من البرنامج أو ايضاح سير عمل البرنامج أو اكتشاف الاخطاء المستقبلية اذا ما وجدت . والان دعنا نلقي نظرة على المخطط الانسيابي النهائي المبين في الشكل (8)


البداية
ادخل عدد الطلبة
T
صفر المعدادات
K, F, M, P
ادخل
اسم الطالب NS
صفر المعداد I
I = I + 1
ادخل درجة
الطالب X
فوق 50
هل الدرجة تحت 50 ؟
نعم تحت 50
J = J + 1
هل I ⩾ 10
كلا
هل J ⩾ 2
نعم
K = K + 1
كلا
هل J ⩾ 0
نعم
اكتب اسم الطالب
النتيجة : راسب
F = F + 1
اكتب اسم الطالب
النتيجة : ناجح
P = P + 1
اكتب اسم الطالب ،
النتيجة : مكمل
M = M + 1
كلا
هل K ⩾ T
نعم
اكتب عدد الناجحين
والمكملين والراسبين
النهاية

الشكل (8)

وهكذا يتبين من هذا المخطط اذا ماقورن مع البرنامج الذي سبق ان شرحناه في الفقرة 25 التوافق التام بين عناصر كل منهما . فجملتي الادخال والاخراج تقابلان رمز الادخال والاخراج وجملة اذا تقابل رمز المقارنة وجمل الاحلال الحسابي تقابل رموز المعالجة .

إن الصعوبة التي ربما لاقيتها في هذا المثال الصعب نوعاً ما لهذه المرحلة ستتذلل اذا مارجعنا الى هذا المثال في المستقبل لتبسيطه اكثر . وحتى لو لم تتقن كافة الجوانب المحيطة بهذا المثال فنقترح مراجعته مرة اخرى على الاقل ثم المضي الى الامثلة الآتية التي هي اسهل نوعاً ما . ثم نقترح فيما بعد العودة الى هذا المثال ومراجعته لتمام اتقانه .

مثال (2)

لنحاول برمجة تسعيرة الطاقة الكهربائية السارية المفعول في القطر الآن . يبلغ سعر وحدة الطاقة الكهربائية (الكيلو واط ساعة) عشرة فلوس اذا كان الاستهلاك اليومي لا يزيد عن 12 وحدة وعشرين فلساً اذا كان بين 12,30 وحدة ، وثلاثين فلساً اذا زاد عن 30 وحدة في اليوم .

اذا عرفت عدد وحدات الطاقة الكهربائية المستهلكة خلال فترة محددة، احسب اجور تلك الطاقة وفق القاعدة اعلاه .

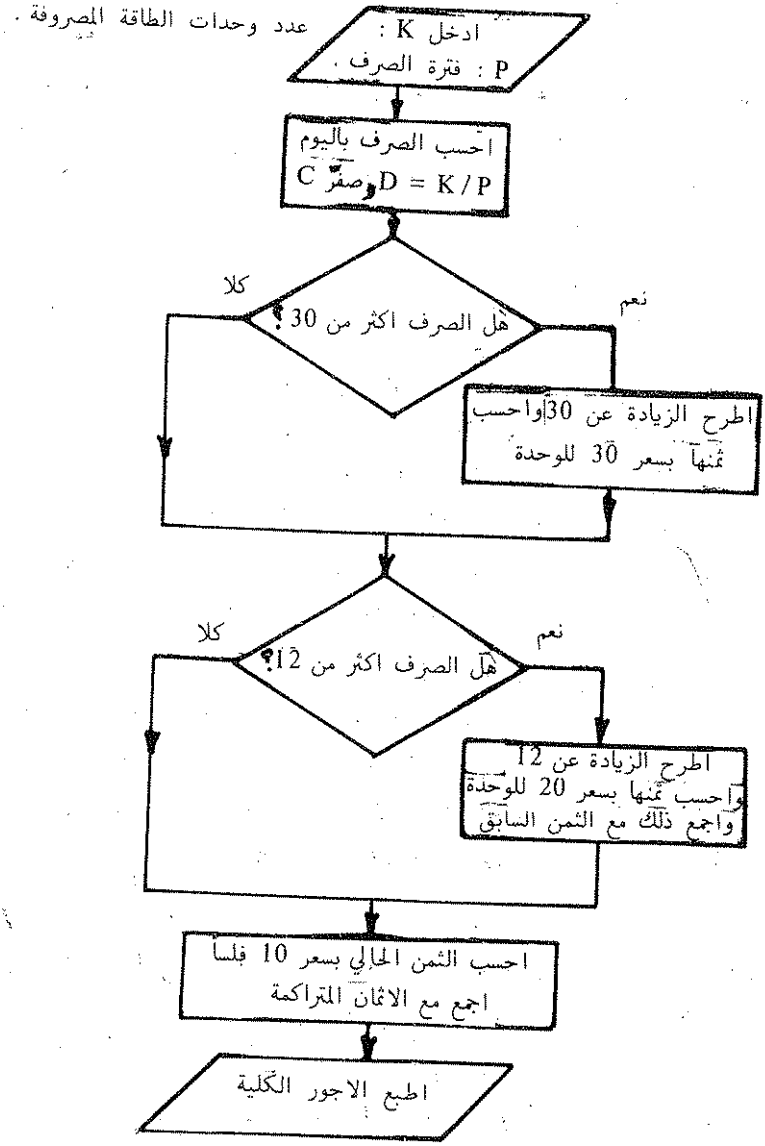
الحل : لنحاول اولاً رسم المخطط الانسيابي للمسألة ، لنفرض ان عدد الوحدات من الطاقة الكهربائية المصروفة تساوي K خلال فترة P يوماً .

فيكون معدل الاستهلاك اليومي . $D = K / P$

لنفرض ان اجور الطاقة الكلية ستكون C . لذلك نفرض اولاً ان $C = 0$ قبل البدء بالمسألة . ان اول تدقيق على الصرف اليومي يتم لمعرفة هل ان D اكبر من 30 أم لا ؟

فاذا كانت اكبر من 30 نطرح 30 من D فيكون ذلك الرقم هو الاستهلاك بسعر 30 فلساً والذي عند ضربه به يعطي اول قيمة للثمن C . ونجري الامر نفسه بالنسبة للمرحلة الثانية (السعر 20 فلساً) .

وبعد ذلك نحسب ماتبقى بسعر 10 فلوس ، وهكذا نكون قد حسبنا كلفة الطاقة الكهربائية ليوم واحد ، والتي عند ضربها في عدد الايام سنحصل على قيمة المبلغ الكلي الواجب دفعه .


ادخل K : عدد وحدات الطاقة المصروفة .
P : فترة الصرف .
احسب الصرف باليوم D = K / P وصفّر C
هل الصرف اكثر من 30 ؟
كلا
نعم
اطرح الزيادة عن 30 واحسب ثمنها بسعر 30 للوحدة
هل الصرف اكثر من 12 ؟
كلا
نعم
اطرح الزيادة عن 12 واحسب ثمنها بسعر 20 للوحدة واجمع ذلك مع الثمن السابق
احسب الثمن الحالي بسعر 10 فلساً اجمع مع الاثمان المتراكمة
اطبع الاجور الكلية

الشكل (9)

والآن لنحاول ترجمة ذلك الى برنامج بلغة بيسيك .

```
10      INPUT   K,P
20      C = 0
30      D = K / P
40      IF D >= 30      THEN 90
50      IF   D >= 12      THEN 120
60      C = C + D * 10
70      PRINT K , "KWH IN" , P, "DAYS; PAYMENT =" C
        "FILS"
80      STOP
90      C = C + (D-30) * 30
100     D = 30
100     GO TO   50
120     C = C + (D - 12) * 20
130     D = 12
140     GO TO   60
```

ماذا تلاحظ في هذا البرنامج ؟

الجمل 30,20,10 سهلة فهي جمل ادخال وتصفير ابتدائي وحساب معدل الصرف باليوم .

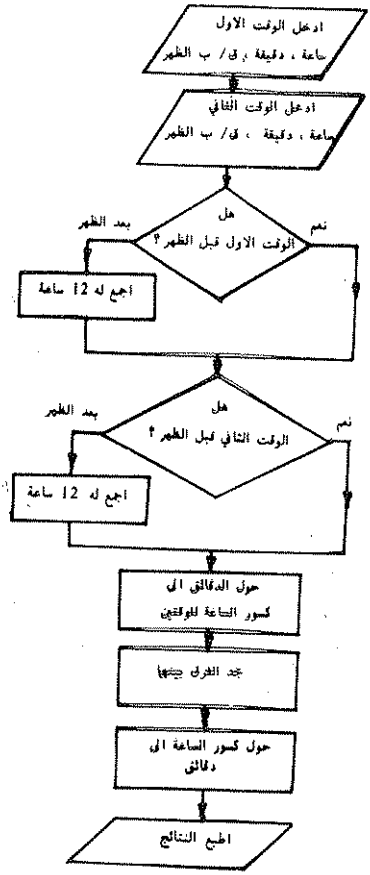
الجملة 40 تقوم بالمقارنة ، هل معدل الصرف اليومي اكثر من 30 ؟ فان كان كذلك انتقل التنفيذ الى الجملة 90 والتي تحسب الكلفة لما يزيد عن 30 وحدة لسعر 30 فلساً . فهي تحسب كلفة ذلك وتجمعه الى C (التي كانت اصلاً صفراً) وبعد ذلك تحذف الزيادة عن 30 وحدة اي تصبح قيمة D ثلاثين وحدة فقط ثم تلتقي مع الجملة 50 التي سيصل اليها التنفيذ حتى ولو كان الصرف اقل من 30 وحدة باليوم .

في الجملة 50 يتم التأكد هل ان الصرف اكثر من 12 وحدة ام اقل . فان كان اقل انتقل الى الجملة 60 والا انتقل الى الجملة 120 التي تحسب كلفة الوحدات التي تزيد عن 12 وحدة (تقع بين 12 ,30) بسعر 20 فلساً وتجمعه لما سبق ان تجمع من اجور في المتغير C . وبعد ذلك تعدل ماتبقى من صرف ليكون 12 وحدة ومن ثم يحسب بسعر 10 فلساً وفق الجملة 60 . وبعد ذلك يتم الطبع والايقاف في الجملتين 70 ,80 .

مثال (3)

كثيراً مانصادف محاولة ايجاد الفرق بين وقتين محددين بالساعات والدقائق مثلاً كم فرق الوقت بين الساعة 7/45 (سبعة وخمس واربعون دقيقة) صباحاً و 3/15 (الثالثة وخمس عشرة دقيقة) بعد الظهر؟ كيف يمكن أن يتم ذلك باستخدام الحاسبة الالكترونية؟

الحل : لنحاول رسم المخطط الانسيابي أولاً والذي يجب أن يحوي كافة الاحتمالات : أن يكون الوقتان كلاهما قبل الظهر او كلاهما بعد الظهر او احدهما قبل الظهر والثاني بعد الظهر او الاول بعد الظهر والثاني قبل ظهر اليوم التالي (للسهولة سنهمل الحالة الاخيرة فقط) ثم نقوم بتحويل الدقائق الى كسور من الساعة وبعد اجراء عملية الطرح (الفرق بين الوقتين) نعيد تحويل كسور الساعة الى دقائق .



الشكل (10)

```
10    INPUT "TIME 1"; H1, M1
15    INPUT "AM OR PM?"; A$
20    INPUT "TIME 2"; H2, M2
25    INPUT "AM OR PM?"; B$
30    IF A$ = "PM" THEN 120
40    IF B$ = "PM" THEN 140
50    H1 = H1 + M1 / 60
60    H2 = H2 + M2 / 60
70    H3 = H2 - H1
80    M3 = INT ((H3 - INT (H3)) * 60)
90    H3 = INT (H3)
100   PRINT "DIFFERENCES IN TIME IS," H3,
            "HOURS", D3, "MINUTES"
110   STOP
120   H1 = H1 + 12
130   GO TO 40
140   H2 = H2 + 12
150   GO TO 50
```

في الجمل 10 و 15 و 20 و 25 قمنا بادخال البيانات المتعلقة بالوقتين . لاحظ أن ادخال الوقت يتم بادخال الساعة ثم الدقيقة ثم يسأل البرنامج هل الوقت قبل الظهر AM أم بعد الظهر PM ؟ فندخل احدهما . اما البرنامج نفسه فيستلم الاجابة على اساس انها متغير حرفي هو A$ للوقت الاول و B$ للوقت الثاني . كما يلاحظ أن المتغيرات الممثلة للساعات والدقائق قد أعطيت بشكل متغير تتكون تسميته من حرف ورقم . وهذا ممكن رغم إننا لم نصادف ذلك من قبل . ويعتبر المتغير H1 مختلف تماماً عن المتغير H2 وكأنه ليس هناك اية علاقة بينهما .

أما الجملتان 30 و 40 فتقومان بمقارنة المتغيرين الحرفيين A$ و B$ مع الصيغة الحرفية "PM" فإن كان الوقت بعد الظهر اضيف اثنا عشر ساعة للساعات وفق الجملتين 120 و 140 ومن ثم يعود التنفيذ الى سير البرنامج العام حسب الجملتين 130 و 150 .

في الجملتين 50 و 60 يتم تحويل الدقائق الى اجزاء الساعة وذلك بالقسمة على 60..ثم يجمع ذلك الى الساعات H1 و H2 .

والآن دعنا نفحص الجملتين 80 و 90 . إنها تحتويان دالة جديدة لم يسبق لنا التعرف عليها هي 1NT .

إن دالة 1NT كغيرها من الدوال المكتبية (تشبه SQR التي مر ذكرها) حيث يوضع مايراد ايجاد دالته بين قوسين وتقوم الدالة بايجاد الجزء الصحيح من ذلك العدد (اي تهمل الكسر العشري) . وعلى هذا فأن دالة العدد 10.5 هي عشرة أي 1NT (10.5) = 10 فقط .

وهكذا فإن الجملة 80 تقوم باستحصال الدالة التي تعطي الرقم الصحيح ويطرح منه العدد الممثل للساعة (معه كسور الساعة اي الدقائق كجزء مئوي من الساعة) . والفرق بينهما هو الكسر فقط بطبيعة الحال . ويتم التحويل الى الدقائق بالضرب في 60 فيكون المتغير M3 عبارة عن الدقائق والتي يمكن ان تحوي كسور الدقيقة .

لذلك نأخذ الجزء الصحيح منها فقط . اما المتغير H3 فيمثل الفرق بين الوقتين مأخوذاً منه جزؤه الصحيح اي الساعات فقط .

وهكذا تطبع النتيجة وفق الجملة 100 لتبيان الفرق بالساعات والدقائق .

28 تمارين

1- اكتب برنامجاً يقوم بطبع جدول يحول الانجات الى سنتمترات بحيث يحتوي على قائمة من 1-20 إنجاً وما يقابلها بالسنتمترات

2- حوّر البرنامج الذي تم ايضاحه في المثال 3 لكي يقبل حالة كون الوقت الاول واقعاً في يوم سابق وان يكون الوقت الثاني تقديره اقل من قيمة الوقت الاول حيث يقوم البرنامج باعطاء سؤال للتأكد من أن الوقت الاول هو في يوم سابق وعندها يقوم بمعالجة هذه الحالة . ارسم المخطط الانسيابي لذلك اولاً ثم اكتب البرنامج الجديد .

3- اكتب برنامجاً يستقبل طول ضلع مكعب صلد وكثافة مادته ويقوم باحتساب حجمه ووزنه . اذا وضع المكعب في الماء (كثافته = 1) بصورة افقية ، هل يطفو ام يغطس؟ واذا طفى كم يكون ارتفاع الجزء الظاهر فوق سطح الماء ؟ ارسم مخططاً انسيابياً لذلك ثم اكتب البرنامج .

29 الحلقات : جملة من --- كرر FOR --- NEXT

تعرفنا في الفقرات الاخيرة على اسلوب تكرار جزء من البرنامج بشكل غير مشروط كما في جملة اقصد GO TO أو بشكل متحكم به من خلال جملة اذا IF--THEN وفي كلتا الحالتين يتكرر الجزء من البرنامج الى أن يتوقف بشكل اضطراري من خلال نفاذ البيانات او بشكل اعتيادي عند تحقق شروط المقارنة في جملة اذا والتي استخدمنا فيها معداداً (متغير) يقوم بحساب عدد الحلقات الدورانية الى أن يصل عددها الحد المطلوب .

ونظراً لكثرة الحاجة اثناء البرمجة الى تكرار اجزاء من البرنامج عدداً معيناً من المرات فإن جملة من .. كرر FOR..NEXT توفر لنا الوسيلة السهلة والواضحة لهذا الغرض .

حينما يراد تكرار حلقة معينة (جزء من برنامج) يوضع في بداية الحلقة جملة وفي نهايتها جملة . جملة البداية تحتوي على تسمية المتغير المحجوز داخل الحلقة وعلى اول قيمة له واخر قيمة له . ويمكن أن يحتوي على مقدار الخطوة التي يجب أن تقفز قيمته بها اذا كانت تلك القيمة لا تساوي واحداً .
اما جملة النهاية فتقوم بطلب تكرار المتغير بقيمة اخرى وذلك على النحو الآتي :
لنفرض أن المطلوب قراءة عدد من البيانات وطباعتها وليكن عشر قراءات :

```
10      FOR     I = 1     TO     10
20      INPUT     X
30      PRINT     I , X
40      NEXT      I
```

في هذه الحلقة يتم بواسطة الجملة 10 تحديد بداية الحلقة بكلمة FOR ويليها اسم المتغير ثم علامة مساواة ثم اول قيمة للمتغير . وبعد ذلك كلمة : الى TO ويليها الحد الاعلى لقيمة المتغير I اما الجملة الاخيرة في الحلقة 40 فتكون صيغتها : كرر NEXT يليها اسم المتغير

وبين هاتين الجملتين نضع ما نحتاج من جمل يراد تكرارها . وقد وضعنا هنا جملتين : الاولى لادخال بيانات X وفي الثانية طباعة قيمتها (X) مع تسلسلها . لاحظ اننا استخدمنا المتغير I كدليل على تسلسل تلك البيانات لان قيمته كانت فعلاً متسلسلة من 1 الى 10 .

ومثل هذه الحالات كثيراً ما نصادفها في المسائل الاعتيادية . ويمكن أن تكون اول قيمة أو اخر قيمة للدليل عدداً ثابتاً كما في المثال السابق ويمكن أن تكون متغيراً

سبق وان حددت قيمته كما في حالة مثال درجات الطلبة السابق ذكره . وفيما يأتي اعادة لذلك البرنامج باستخدام ما تعلمناه الان من صيغة FOR .. NEXT

```
60      INPUT     T
70      P = 0
80      M = 0
90      F = 0
101     FOR     K = 1     T Θ     T
120     J = 0
130     INPUT     N$
131     FOR     I = 1     TO     10
140     INPUT     X
150     IF     X >= 50     THEN     171
155     J = J + 1
171     NEXT     I
180     IF     J > 2     THEN     250
190     IF     J > 0 THEN     220
200     PRINT     N$ , "PASS"
205     P = P + 1
210     GΘ     T Θ     271
220     PRINT     N$ , "MAKE UP IN", J, "SUBJECTS"
230     GΘ     T Θ     271
250     PRINT     N$ , "FAILED IN", J, "SUBJECTS"
255     F = F + 1
271     NEXT     K
275     PRINT "PASS", P, "MAKE UP", M,
        "FAIL", F, "TOTAL", P + F + M
280     STOP
```

وبمقارنة هذا البرنامج مع البرنامج السابق نلاحظ ما يأتي :

أ) استعضنا عن الجمل 100 و 260 و 270 بالجملتين 101 و 271 واللتان تحددان بداية ونهاية الحلقة وفيها اسم المتغير السابق نفسه (K) .

ب) استعضنا عن الجمل 110 و 160 و 170 بالجملتين 131 و 171 باستخدام المتغير I نفسه . لاحظ أن بداية هذه الحلقة لم يوضع عند الجملة 110 بل

بعد الجملة 130 لانها يجب أن تكون بعد قراءة الاسم N$ وتصفير دروس الرسوب المعطاة بالجملة 120 .

جـ) تم اجراء بعض التعديلات التي اقتضتها التغييرات السابقة في ارقام الجمل المقصودة كما في 210 و 230 .

د) لاحظ حيز عمل الحلقتين المؤشر بالقوسين على يمين البرنامج فهما حلقتان احداهما داخل الاخرى .

هـ) لاحظ أن بداية الحلقة الاولى : الجملة 101 قد بدات من قيمة للمتغير = I وانتهت بالقيمة T . وهي متغير ايضاً لكن سبق أن حددت قيمته من خلال الجملة 60 وهذا ممكن وصحيح ومفيد .

وهكذا من هذا المثال يمكننا أن نرى كيف أن البرنامج قد تم تبسيطه واصبح اكثر وضوحاً حيث تم الاستغناء عن المعداد وعن عمليات المقارنة التي يقتضيها انهاء حلقات المعداد . والان نود أن نضيف امراً جديداً وهو الخطوات التي تقفز بها قيمة الدليل وذلك باستخدام كلمة خطوة : STEP وذلك بواسطة بعض الامثلة :

30 ـ امثلة على الحلقات

مثال 1 : جد مجموع الاعداد التي تقبل القسمة على 7 وتقع بين 100 و 200

الحل : ان اول عدد يقبل القسمة على 7 بعد 100 هو 105 لذلك سنبدأ به . ولكن قبل ذلك نحاول حجز اسم لمتغير لوضع المجموع فيه وليكن هذا الاسم S . نبدأ اولاً بتصفيره .

```
10      S = O
20      FOR     I = 105     T O     200     STEP 7
30      S = S + I
40      NEXT    I
50      PRINT   "SUM", S
```

لقد بدأنا باول عدد يقبل القسمة على 7 وجعلنا خطوات قفز قيمة I سبعة سبعة الى أن نصل الى الحد الاعلى وهو 200 . لاحظ أن هذا الحد 200 نفسه لا يقبل القسمة على 7 . لذلك فإن اخر عدد محسوب (يقبل القسمة على 7) هو 196 . واذا ما اضيف له 7 فسيخرج الحد عن 200 (يصبح 203) . لذلك تتوقف الحلقة عند 196 رغم أن العدد المكتوب هو 200 . ويمكن التأكد من ذلك بكتابة كافة هذه الاعداد مثلاً من خلال جملة

```
25      PRINT   I
```

وهكذا تلاحظ اننا تعاملنا مع المتغير I بصفته متغيراً داخلاً في الجملة 25 لطبع قيمته أو ضمن عمليات الجمع في الجملة 30 وذلك مفيد وممكن . الا اننا يجب أن نحذر بعدم تغيير قيمته من قبلنا بواسطة جملة احلال . فإن جملة مثل :

35 I = I + 1

خطأ ولا يجوز كتابتها فطالما اننا قد حجزنا هذا المتغير لكي تتحكم بقيمته جملة من ... كرر فلا يحق لنا التلاعب بقيمته من قبلنا بعد ذلك .

ونود أن نشير هنا الى أن قيمة دليل الخطوة يمكن أن تكون موجبة أو سالبة . فاذا كانت اخر قيمة لدليل الحلقة اكبر من اولها وجب أن تكون قيمة الخطوة موجبة . أما اذا كانت القيمة الثانية (الاخيرة) اصغر من الاولى فإن الخطوة يجب أن تكون قيمتها سالبة لغرض الوصول نحو القيمة النهائية خطوة خطوة . فمثلاً

50 FOR I = 70 T Θ 50 STEP - 2

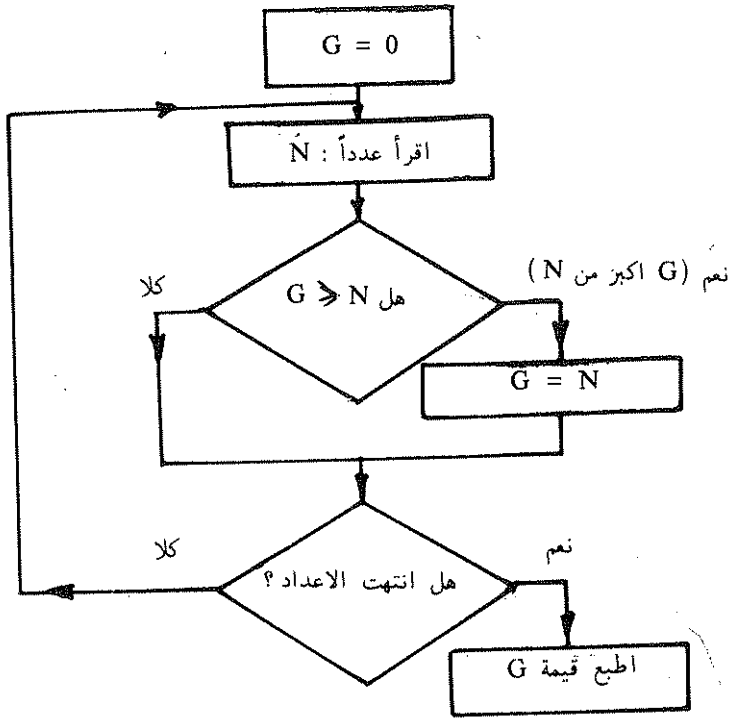
هي جملة صحيحة وتنفذ ما يطلب منها من خطوات متدرجة مبتدئة بـ I = 70 ثم I = 68 وهكذا

وبالطبع يجب الحذر من عدم اعطاء قيمة صفر للخطوة لانها ستصبح كمن يراوح في محله غير متقدم الى الامام ولا الى الوراء وستبقى الحلقة تدور بلا فائدة الى مالا نهاية ، الا اذا اوقفت بواسطة مفتاح الايقاف STOP

مثال 2 : كيف تقوم بايجاد اكبر عدد من بين مجموعة اعداد ، لنفرض أن عددها 100 عدد ؟

الحل : لنحاول اولاً أن نرسم المخطط الانسيابي للمسألة . افرض أن اكبر عدد هو G . افرض اولاً أن قيمته = صفراً .

اقرأ العدد الاول ثم قارن قيمته مع G ، فاذا كان G اكبر من ذلك العدد ابقيناه على حاله . اما اذا كان G اصغر من ذلك العدد غيرنا قيمة G لتصبح مساوية لذلك العدد . وهكذا نستمر حتى ننتهي من الاعداد المائة .



الشكل (11)

والآن لنحاول كتابة البرنامج :

```
10    G = O
20    FOR   I = 1   T Θ   100
30    1NPUT   N
40    1F   G > N   THEN   60
50    G = N
60    NEXT   I
70    PRINT "THE   GREATEST   NUMBER =" ,G
```

ماذا يجب أن يضاف للبرنامج لكي يحسب لنا اصغر عدد بين المجموعة ايضاً ؟
الجواب سهل :

```
15    S = 1000000
```

احجز S لخزن اصغر عدد واعطه قيمة كبيرة جداً مثلاً 1000000
ثم اضف الجملتين الآتيتين لايجاد أصغر عدد

```
32    IF   S < N   THEN   40
34    S = N
```

ثم جملة للكتابة :

```
75    PRINT  "THE  SMALLEST  NUMBER =", S
```

وهكذا فإن الحلقة نفسها (لكل . . كرر) استخدمناها لغرض ايجاد اكبر عدد وأصغر عدد في الوقت نفسه . وهذا بالطبع لا يمنع من استخدامها لاغراض أخرى اذا احتجنا لذلك ونفذنا ما نريد بشكل صحيح .

مثال (3) : لنحاول التعويض في معادلة من الدرجة الثالثة بهدف رسمها . مثلاً المعادلة :

$$y = X^3 - 5X + 1$$

لنكتب برنامجاً صغيراً يقوم بذلك مثلاً .

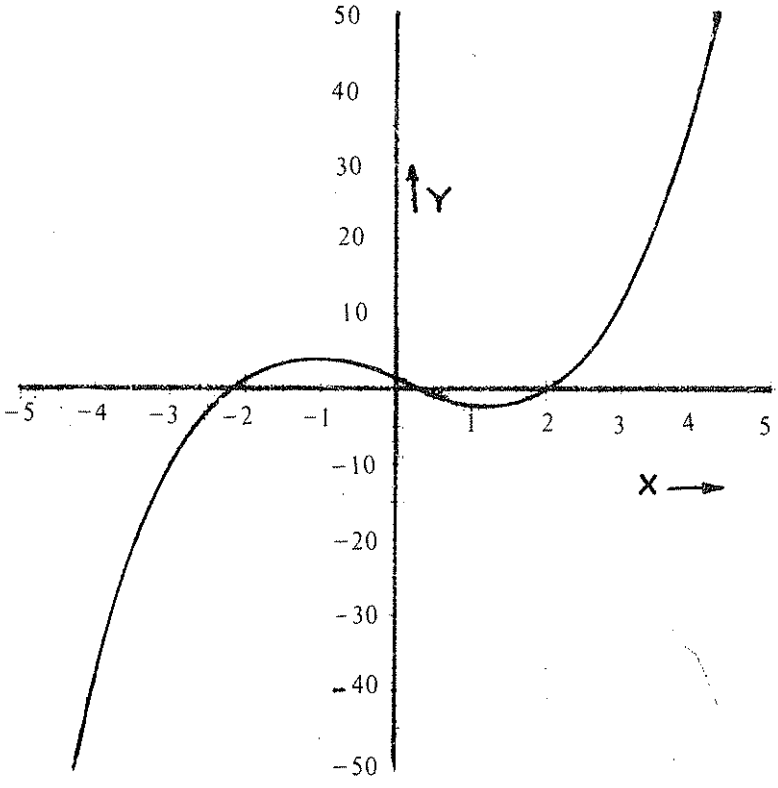
```
 5    INPUT    A,B,C
10    FOR    X = A    T O    B    STEP    C
20    Y = X ^3 - 5 *X + 1
30    PRINT    X, Y
40    NEXT    X
```

هذا البرنامج يعوض قيمة X في المعادلة اعتباراً من A ثم يضيف خطوات تعادل C لكل مرة الى أن تصل قيمة B ويقوم كل مرة بطبع قيمتي X و Y وهكذا ستحصل على عدد من النقاط بحيث تكون كافية لرسم المنحني Y مع X كما مبين في الشكل (12) . لنفرض اننا حددنا A = −5 و B = 5 و C = 1 . وبالطبع فإن الحدود من 5 − الى 5 + هي ارقام تخمينية اولية . واذا ما اعطينا هذه القيم للبرنامج السابق فإننا سنحصل على قيم Y المبينة في الجدول (1) . ولكن لنفرض إن المطلوب هو ايجاد حلول المعادلة (Y = 0) فهذه معادلة من الدرجة الثالثة لذلك لها ثلاث حلول ولا نعرف لها دستوراً للحل . فماذا نفعل؟

إن البرنامج السابق يمكن أن يساعدنا في الحل . فبعد أن حصلنا على الجدول (1) لقيم X بين 5 − و 5 + استطعنا أن نرسم الشكل (12) بصورة تخمينية تقريبية . ويتضح وجود ثلاثة حلول (نقاط تقاطع مع محور السينات) وهي تقع بين :

الاول بين 3 − و 2 − والثاني بين 0 و 1 والثالث بين 2 و 3 لذلك نعيد المحاولة بتشغيل البرنامج نفسه ولكن باعطاء بيانات أكثر تحديداً بتقليل قيمة الخطوة C من 1 الى 0.1 مثلاً . فنحصل على الجداول (2) و (3) و (4) . والتي منها يتضح إن الجذور الثلاثة كالآتي :

الاول بين 2.3 − و 2.4 − والثاني بين 0.2 و 0.3 والثالث بين 2.1 و 2.2 واذا ما اردنا زيادة الدقة نعيد البرنامج ثلاث مرات اخرى ولكن هذه المرة بقيمة C



الشكل (12)

أصغر مما سبق مثلاً 0.01 فنحصل على الجداول الثلاثة الجديدة (5) و (6) و (7) والتي تبين إن الجذور الثلاثة تساوي تقريباً 2.33- و 0.2 و 2.13 .

وهكذا استخدمنا الحاسبة الالكترونية باستخدام البرنامج نفسه لحل هذه المعادلة تدريجياً خطوة خطوة الى أن توصلنا الى اقرب ما يمكن من الحلول مقرّبة الى مرتبتين يمين الفارزة .

والسؤال الذي يتبادر للذهن الآن هو : الا يمكن أن تقوم الحاسبة نفسها بالاهتداء الى هذه الحلول دون تدخل من قبلنا ؟ الجواب نعم . وهناك اساليب لذلك . ولكن ذلك خارج موضوعنا هنا ويحتاج الى خبرة في موضوع التحليل العددي .

جدول 1

X	Y
−5	−99
−4	−43
−3	−11
−2	3
−1	5
0	1
1	−3
2	−1
3	13
4	45
5	101

جدول 2

X	Y
−3	−11
−2.9	−8.889
−2.8	−6.952
−2.7	−5.183
−2.6	−3.57601
−2.5	−2.12501
−2.4	−.824007
−2.3	.332994
−2.2	1.35199
−2.1	2.23899
−2.	2.99999

جدول 3

X	Y
0	1.00001
0.1	.501009
0.2	8.00920E-03
0.3	−.472991
0.4	−.935992
0.5	−1.37499
0.6	−1.78399
0.7	−2.15699
0.8	−2.48799
0.9	−2.771
1.0	−3.

جدول 4

X	Y
2.	−1.00002
2.1	−.239021
2.2	.647976
2.3	1.66697
2.4	2.82397
2.5	4.12496
2.6	5.57595
2.7	7.18295
2.8	8.95194
2.9	10.8889
3.	12.9999

جدول 5

X	Y
−2.4	−.824001
−2.39	−.701918
−2.38	−.581268
−2.37	−.462048
−2.36	−.344248
−2.35	−.227863
−2.34	−.112888
−2.33	6.79016E−04
−2.32	.11285
−2.31	.223629

جدول 6

X	Y
.2	8.00002E−03
.21	−4.07391E−02
.22	−8.93519E−02
.23	−.137833
.24	−.186176
.25	−.234375
.26	−.282424
.27	−.330317
.28	−.378048
.29	−.425611

جدول 7

X	Y
2.1	−.239
2.11	−.156069
2.12	−7.18689E-02
2.13	1.36013E-02
2.14	.100349
2.15	.188383
2.16	.277706
2.17	.688328
2.18	.460249
2.19	.553476

31 تداخل الحلقات

لاحظنا في برنامج درجات الطلبة أن هناك حلقتين احداهما داخل الاخرى وقد استعملنا لهما دليلين مختلفين (الخارجية K والداخلية I) كل من هاتين الحلقتين طبعاً تبتديء بجملة FOR وتنتهي بجملة NEXT . الحلقة الخارجية تنفذ بعدد الطلبة مثلاً اربعين مرة والداخلية تنفذ عشر مرات لكل طالب أي 400 مرة لكل الطلبة .

وهكذا كان البرنامج ناجحاً للمهمة التي كتب من اجلها . ولكن ما هي القواعد أو المحددات التي ينبغي مراعاتها عندما نصادف حلقات متعددة في برنامج واحد ؟ لنبدأ بذكر كافة انواع العلاقات بين الحلقات :

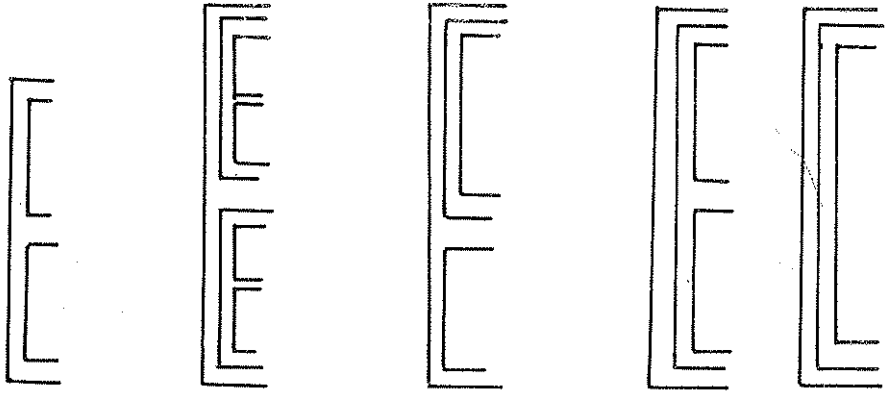
أ) الحلقات المتتالية : وذلك بأن نفرغ من حلقة ثم ندخل الحلقة الثانية بشكل تعاقبي . وهذا ممكن ولا يسبب اية صعوبة . ويمكن في هذه الحالة استخدام اسم الدليل نفسه للحلقتين لعدم وجود ارتباط بينهما : مثلاً

```
10 FOR    I = 1    T Θ    10
20 INPUT    X
30 PRINT    X, X ^2
40 NEXT    I
50 FOR    I = 1    T Θ    20
60 PRINT    I, I ^3
70 NEXT    I
```

فالحلقة الاولى تقرأ عشرة ارقام وتطبع مربعاتها والثانية تطبع الاعداد من 1 الى 20 مع مكعباتها . وقد استخدم الدليل I لكلا الحلقتين .

ب) الحلقات المتداخلة تماماً : وذلك كما سبق أن بينا في مثال درجات الطلبة حيث تكون الحلقتان متداخلتين أحداهما تضم الاخرى تماماً . ويجب استخدام دليلين مختلفين تماماً لهما .

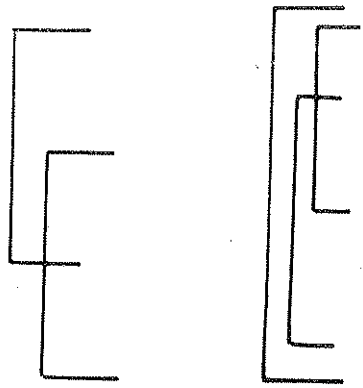
جـ) الحلقات المركبة من التداخل والتتالي : فمثلاً أن كان هناك حلقة كبيرة خارجية دليلها K وفيها حلقتان تنتهي الاولى ثم تبتديء الثانية فيمكن استخدام المتغير I نفسه كدليل لكلا هاتين الحلقتين او استخدام متغير مختلف لكل منها . وهناك اشكال عديدة لهذا النوع من الحلقات منها ما مبين في الشكل (13)



الشكل (13)

ويلاحظ في كافة هذه الانواع من الحلقات اننا لا يمكن أن ننهي الحلقة الخارجية الا اذا تم انهاء الحلقة او الحلقات الموجودة بداخلها اولاً .
فالصيغة المتقاطعة في الحلقات مرفوضة تماماً ولايجوز اجراؤها ابداً لانها غير منطقية وتربك مفهوم البرنامج . ويبين الشكل (14) نماذج من الحلقات المتقاطعة التي لا يجوز استخدامها

أما الامور الاخرى التي تجب مراعاتها في الحلقات فهي عملية الدخول الى الحلقة والخروج منها .



الشكل (14) حلقات متقاطعة لا يجوز استخدامها

فالدخول الطبيعي الى الحلقة يكون من خلال بابها الشرعي وهو جملة FOR والخروج يكون من نهايتها الصحيحة جملة NEXT . لكن في بعض الاحيان يصادف اثناء البرنامج وجود حاجة الى الانتقال الى جملة تقع خارج حيز عمل الحلقة لاجراء بعض الحسابات او المعالجات ثم العودة مرة اخرى الى داخل تلك الحلقة . أن هذا الاجراء صحيح وليس هناك غبار عليه .

كما أن هناك حالات تستدعي الدخول الى الحلقة مؤقتاً لاجراء بعض المعالجات والخروج منها دون استخدام لرقم الدليل الذي تستخدمه الحلقة . فهذا الاجراء يتم كأنه اجراء خارجي لاعلاقة للحلقة به وليس فيه أي عمل يناقض قواعد الحلقات . الا انه يجب الحذر في كلا هذين الانتقالين داخل وخارج الحلقات لئلا يتم ارتكاب ما يناقض تلك القواعد .

لنأخذ مثالاً على ذلك :

```
10    INPUT    X
20    IF    X > 5    THEN    70
30    FOR    I = 1    T O    10
40    PRINT    I ^2
50    Y = X * I
70    Z = X ^2
80    PRINT    Z
90    IF X > 5    THEN    110
100   NEXT    I
110   PRINT    X
```

ماذا تلاحظ ؟ اذا كانت قيمة X اقل من 5 او تساوي 5 فإن الجملتين 20 و 90 لا فائدة لهما . لذلك فإن البرنامج يكون صحيحاً اذ يحتوي على حلقة واحدة ذات شكل اعتيادي .

اما اذا كانت قيمة X اكبر من 5 فإن الانتقال سيتم داخل الحلقة مباشرة الى الجملة 70 والتي بواسطتها يتم حساب قيمة Z التي هي مربع X ثم طباعة ذلك ثم الانتقال خارج الحلقة بواسطة الجملة 90 مباشرة الى الجملة 110 حيث تطبع قيمة X . وهذا البرنامج صحيح ايضاً . وليس فيه اية مخالفة لقواعد الحلقات .

اما لو كانت الجملة 20 بصيغة

```
21    IF X<=5    THEN    50
```

فهذه صيغة خاطئة وستظهر اثارها اثناء عملية تنفيذ البرنامج .
فإن الانتقال قد تم الى الجملة 50 وهذه الجملة تعتمد في متغيراتها على قيمة المتغير I الذي هو دليل الحلقة . فاذا لم نكن قد دخلنا الحلقة من بابها الاعتيادي فمن اين لهذه الجملة المعرفة بقيمة I حيث لم تمر على جملة FOR التي تحدد قيمة الدليل I . وهكذا فإنها ستفرض أن قيمة I تساوي صفراً او اية قيمة سبق أن اعطيت في جمل سابقة اثناء البرنامج (فيما اذا سبق استخدام المتغير I في حلقة او جملة احلال سابقة) بشكل لا علاقة له بهذه الحلقة . لذلك فإن الدخول والخروج من والى الحلقات يستحسن تجنبه قدر الامكان الا اذا اضطررنا اليه وعند ذلك ينبغي الحذر من الوقوع في الخطأ .

32 ـ تمارين

1- اكتب برنامجاً يقوم بايجاد الجذر التربيعي وطباعته للاعداد من 1 الى 10 واحداً واحداً ثم من 10 الى 100 عشرة عشرة ثم من 100 الى 1000 مائة مائة وذلك باستخدام ثلاث حلقات متتالية . كيف يمكن أن يحّور البرنامج للوصول الى الهدف نفسه باستخدام حلقتين متداخلتين ؟

2- اكتب برنامجاً يقوم بحساب عناصر متوالية هندسية حدها الاول 100 وعدد حدودها 10 والنسبة بين كل حدين فيها تساوي 1.1 كما يقوم البرنامج بايجاد مجموعة هذه العناصر .

3- بالرجوع الى المثال (2) في الفقرة (22) والمتعلق بعمل تقويم لشهر معين ، كيف يمكن تحوير ذلك البرنامج لكي يستخدم حلقة FOR .. NEXT ؟ هل يمكن اضافة جزء للبرنامج لكي يقرأ عدد ايام الاشهر الاثني عشر من خلال جملة (READ) مع DATA ويقوم بعمل تقويم سنوي يصلح لأي شهر من اشهر السنة ؟ حاول ذلك .

33 ـ ايعازات تنفيذية اضافية

لقد استخدمنا لحد الآن ايعازين تنفيذيين فقط هما RUN و LIST . هل هناك ايعازات اخرى؟ نعم ..

أ) ايعاز جديد NEW

هذا الايعاز يقوم بالغاء البرنامج الذي سبق أن ادخل الى الحاسبة ويبدأ من جديد . فهو يمحي كافة المتغيرات السابقة . لذلك يجب عدم استخدام هذا الايعاز الا بعد التأكد التام من عدم الحاجة الى ذلك البرنامج ولا حتى الحاجة الى تعديله ليصبح برنامجاً جديداً .

ب) ايعاز داوم CONT

سبق أن شرحنا سابقاً اسلوب ايقاف برنامج بشكل اضطراري من خلال الضغط على مفتاح STOP . ولكن بعد أن اوقفنا هذا البرنامج ثم اردنا أن نعود اليه مرة اخرى فيمكن أن يتم ذلك بواسطة ايعاز CONT فهو يعيد اكمال التنفيذ كما لو انه قد تم ادخال جملة اقصد تعود الى الجملة التي توقف الحل عندها (GO TO..)

جـ) ايعاز ذاتي AUTO

وهو متوفر على حاسبة صخر ولا يتوفر على حاسبة الوركاء ويقوم ذاتياً بتزويد ارقام الاسطر . ويمكننا التحكم بواسطته برقم اول سطر نبدأ به وكذلك مقدار الخطوة بين رقم سطر وآخر مثلاً

AUTO 105 و 5

والتي تعني أن البرنامج سوف يبدأ بالجملة 105 ثم تزداد ارقام الجمل خمسة خمسة . فكلما ادخلت جملة فإن رقم الجملة التالية يظهر ذاتياً على الشاشة . ويمكن الخروج من صيغة الترقيم الذاتي بالضغط على مفتاح السيطرة ومفتاح الايقاف بوقت واحد . وعند ذلك يمكنك بعد ذلك الاستمرار بالبرنامج مدخلاً ارقام الجمل بنفسك .

د) ايعاز احذف DELETE

يقوم هذا الايعاز بحذف جزء من البرنامج وعلى النحو الآتي :

لحذف الجمل من بداية الجملة 60 الى نهاية الجملة 99

```
DELETE    60 - 99
```

لحذف الجمل من بداية الجملة 60 الى آخر البرنامج (فقط حاسبة الوركاء)

```
DELETE    60 -
```

لحذف الجمل من بداية البرنامج الى نهاية الجملة 60

```
DELETE    - 60
```

لحذف الجملة 60 فقط

```
DELETE    60
```

هـ) ايعاز ترقيم RENUM

اذا قمت باجراء تحويرات كثيرة في برنامج ما ووجدت أن الترقيم اصبح مرتبكاً فيمكنك اعادة ترقيم كافة جمل البرنامج ذاتياً بواسطة هذا الايعاز . ويمكن أن يكون الايعاز مطلقاً لكافة جمل البرنامج

```
RENUM
```

وستجد أن البرنامج يقوم حينئذ بتحوير كافة ارقام الجمل الى ارقام جديدة وكذلك كافة الجمل التي تشير اليها مثل GO TO أو IF . . . THEN أيضاً . واذا ما اردت تحوير ارقام بعض الجمل فقط فيمكن ذلك باعطاء رقم الجملة القديمة والرقم الجديد الذي تريد أن تعطيها اياه . وعند ذلك سيتم اعادة ترقيم كافة الجمل اعتباراً من تلك الجملة حتى نهاية البرنامج .

هذا يصلح لكل من حاسبتي صخر والوركاء ، الا انه هناك قابلية اضافية في حاسبة صخر بأنه يكون بامكانك تحديد الخطوة ايضاً بين رقم وآخر من الترقيم الجديد . فمثلاً

```
RENUM     100 , 60 , 5
```

يعني اعادة ترقيم كافة جمل البرنامج من الجملة 60 الى نهاية البرنامج بحيث تصبح 100 , 105 , 110 وهكذا . اما على حاسبة الوركاء فلا يمكن سوى

```
RENUM     100 , 60
```

34 ايعازات جمل برمجية اضافية

أ) جملة شرح REM

هذه الجملة تأخذ رقماً بين الجمل ولا تفعل شيئاً اثناء التنفيذ . إنها جملة تبتديء بهذه الاحرف الثلاثة ويمكنك استخدام سطرها لكتابة ملاحظاتك مهما كانت تلك الملاحظات . وهي مفيدة لشرح اجزاء من البرنامج ليسهل الرجوع اليها في المستقبل . ولهذه الجملة صيغة بسيطة واحدة في حاسبة صخر .

اما في حاسبة الوركاء فيمكن استخدامها مع متغير حرفي ضمن البرنامج بحيث يطبع تلك الملاحظة المعطاة بواسطة المتغير الحرفي اثناء الوصول الى ذلك السطر .

مثال :

```
10      A$ = "DATA ARE WRONG"
20      INPUT K
30      IF K > 100 THEN 80

80      REM A$
```

فاذا كانت قيمة K أكبر من 100 انتقل الحل الى الجملة 80 لكتابة الملاحظة التي تعني أن البيانات خاطئة .

ب) جملة النهاية « تم » END

تقوم هذه الجملة بانهاء البرنامج ويمكن أن يحتوي البرنامج الواحد على اكثر من جملة END واحدة وذلك في مناطق انهاء البرنامج .
ورغم أن البرنامج الذي لا ينتهي بجملة END يقف بشكل صحيح ، الا أن السجلات التي يفتحها البرنامج لا تغلق الا بجملة END ، وهذا خارج ما سبق أن شرحناه .

35 ـ صيغ برمجية اخرى

أ) استخدام اضافي لجملة IF.. THEN

سبق أن شرحنا الاستخدام الاعتيادي لجملة اذا .. اذن IF.. THEN حيث عند تحقق جملة المقارنة بعد IF ينتقل التنفيذ الى الجملة التي يعطي رقمها بعد THEN

إن بالامكان استخدام هذه الجملة بحيث بدلاً من الانتقال الى جملة اخرى تكتب جملة كاملة مباشرة بعد THEN . ويمكن أن تكون هذه الجملة جملة ادخال أو اخراج أو احلال حسابي .
مثال

```
10      IF A > 8 INPUT B
20      IF A < 16 THEN B = 118
30      IF A * B > = 25 THEN PRINT A,B
```

وهكذا يمكن بهذه الوسيلة اختصار جملتين أو اكثر في جملة IF واحدة .
وينطبق هذا على حاسبتي صخر والوركاء . الا أن في حاسبة صخر هناك خاصية اضافية مفيدة جداً وهي امكانية استخدام ELSE والتي تعني (بعكسه) . فمثلاً :

```
15      IF A > 8 THEN 28 ELSE 17
```

ويعني ذلك اذا تحقق الشرط A > 8 فيتم الانتقال الى الجملة 17 وبعكسه اذا لم يتحقق الشرط ذلك فيتم الانتقال الى الجملة 28 . ويمكن الاستعاضة في هذه الحالة عن رقمي الجملتين 17 و 28 بجملتين كاملتين . مثلاً

```
32      IF A > B THEN PRINT "CORRECT"
        ELSE PRINT "WRONG"
```

وهكذا نكون بهذه الجملة الفريدة قد عوضنا عن اربع جمل هي :

```
10      IF A > B THEN 28
17      PRINT "WRONG"
20      GO TO 30
28      PRINT "CORRECT"
30
```

ب) كتابة جمل متعددة على سطر واحد

بالامكان كتابة اكثر من جملة واحدة على سطر واحد برقم واحد بشرط أن لا يكون هناك حاجة للرجوع الى الجمل الاخرى عدا الاولى .
ويفصل بين هذه الجمل بنقطتين : مثلاً يمكن أن يكتب :

```
10 INPUT A,B : C = A * B : PRINT C
```

فهذه ثلاث جمل هي برنامج كامل ينفذ مرة واحدة ويمكن أن يرجع اليه فقط من خلال رقم الجملة 10 . فلا يمكن الرجوع الى الجملة الوسطية (C = A * B)

مثلاً من خلال جملة اقصد مثلاً الا بالاشارة الى رقم الجملة 10 . وعند ذلك تنفّذ الجمل الثلاثة سويةً .

جـ) حدود الثوابت

المقادير الثابتة بلغة بيسيك يمكن أن تكون اعداداً صحيحة او اعداداً بكسور عشرية ويمكن أن تكون موجبة او سالبة

حدود الاعداد الصحيحة من 32768- الى 32767

العدد المحتوي على فارزة يمكن أن يكون محصوراً بين 10^{63} , 10^{-64}

ويمكن أن يكتب بصيغة الكسر الاعتيادية بشرط أن لا تزيد أرقامه عن 8 أو بصيغة الاسس باستخدام حرف E مثلاً :

6.2 E-2

ويعني ذلك 6.2×10^{-2} أي 0.062

هذا وهناك انواع اخرى من الثوابت هي الثوابت بالنظام السداسي عشر والتي سنؤجل الخوض فيها الآن . وهناك على حاسبة صخر امكانيات واسعة لاستخدام الدقة المضاعفة والنظام الثماني والنظام الثنائي .

اما الثابت الـ في فإن اكبر عدد لحروفه هو 255 حرفاً ويوضع بين اقواس مقلوبة

د) حدود المتغيرات

المتغيرات الرقمية تدعى باسم يبتديء بحرف من الحروف الابجدية ويمكن أن يحتوي على حرفين ثانيهما رقماً من الارقام العشرة 0-9 أو حرفاً من الحروف الابجدية ، مثلاً .

AA, A9, C3, F2, B,H

ويمكن أن تأخذ هذه المتغيرات قيماً ثابتة بحدود الثوابت التي مرت أعلاه . كما يمكن تمييز المتغيرات التي تأخذ قيماً حرفية وذلك باشارة $ في نهايتها ، مثلا

AA$, B$, X$, Y1$

36 الدوال المكتبية

سبق أن ذكرنا دالتين مكتبيتين هما الجذر التربيعي SQR ودالة القيمة الصحيحة INT وسنورد هنا دوالاً اخرى تكثر الحاجة اليها .

أ) دالة القيمة المطلقة ABS

وتعطي هذه الدالة القيمة المطلقة بحذف الاشارة السالبة إن وجدت فإن :

ABS (-5) = ABS (5) = 5

ويمكن أن تكون الكمية بين القوسين ثابتاً او متغيراً او مقداراً حسابياً او جبرياً .

ب) دالة جيب الزاوية SIN

تعطي هذه الدالة جيب الزاوية شرط أن تكون الزاوية مقدرة بالتقدير نصف القطري (Radians) . وتكون بالطبع قيمة الجيب محصورة بين 1- و 1+ .

مثال ذلك :

Y = SIN (x)

أو

Y = SIN (3.1415926 * I / 180)

والاخيرة مفيدة للتحويل من التقدير بالدرجات الى التقدير بالزوايا نصف القطرية (اذا كانت I بالدرجات) .

ويمكن أن تكون الكمية بين القوسين معقدة المقدار او بسيطة حسب الحاجة .

جـ) دالة جيب التمام COS

هذه الدالة تشبه في مواصفاتها دالة الجيب وصيغتها هي COS (x)

د) دالة ظل الزاوية TAN

وتشبه هذه الدالة في مواصفاتها دالة الجيب ايضاً وصيغتها TAN (x) الا أنها محصورة بين اصغر واكبر الارقام التي تتعامل معها الحاسبة أي

10^{63} الى 10^{-64} , 10^{63} - الى 10^{-64} -

اما بقية الدوال المثلثية فيمكن اشتقاقها من هذه الدوال الثلاثة ، مثلاً القاطع هو مقلوب جيب التمام وظل التمام مقلوب الظل .. وهكذا .

هـ) دالة اللوغاريتمات الطبيعية L Θ G

تستخدم هذه الدالة لاستخراج لوغاريتم عدد او مقدار للاساس e

و) دالة الرفع الى اساس اللوغاريتمات الطبيعية EXP

تستخدم هذه الدالة لرفع كمية الى الاساس e . وهي مفيدة في ايجاد دوال القطوع الزائدية مثل sinh , cosh , tanh مثلاً .

TA = (EXP (x) − EXP (−x)) / (EXP (x) + EXP (−x))

على فرض أن المطلوب حساب TA = tanh (x)

ز) الدالة العشوائية RND

تعطي هذه الدالة عدداً عشوائياً (تقوم باختياره لاعلى التعيين وبالطبع وفق روتين معين) وتغذّى الدالة العشوائية عادة بعدد . فاذا كان :

العدد موجباً : تستحدث عدداً عشوائياً من ضمن منظومة الاعداد العشوائية نفسها

العدد صفراً : تعطي العدد العشوائي الذي سبق أن تم توليده (تكرار القيمة)

العدد سالباً : تستحدث عدداً عشوائياً بواسطة منظومة اعداد عشوائية جديدة .

مثلاً لايجاد عشرة اعداد عشوائية بين 100,1 :

```
10      FOR  I = 1 T Θ 10
20      A = RND (1)
30      PRINT INT (100 * A
40      NEXT I
```

ح) قياس الوقت TIME

هذا المتغير TIME محجوز الاسم من قبل الحاسبة وهو ليس دالة كغيره حيث لا يحتاج الى دليل . وهو وسيلة لقياس الزمن على الحاسبة . تزداد قيمة الزمن TIME بمقدار 2 كل فترة $\frac{2}{975}$ من الثانية بالنسبة لحاسبة الوركاء وتتراوح قيمة المتغير بين 32768- ,65535 .

اما على حاسبة صخر فتزداد قيمته كل $\frac{1}{50}$ من الثانية ويعيد الرقم نفسه كل 21.8 دقيقة .

37 برامج تسلية

البرنامج الاول : احزر الرقم الذي تضمره الحاسبة .
في هذا البرنامج تقوم الحاسبة باضمار رقم عشوائي وتحاول أنت اكتشافه باقل عدد من المحاولات .

اولاً ستسألك الحاسبة عن اسمك ثم عن حدود الارقام المسموح لها باختيار الرقم المضمر أي بين الصفر والرقم الذي تزوده انت كحد أعلى . وعندما تضمر الحاسبة الرقم تطلب منك أن تحزره وستجيبك عند محاولتك بأحد ثلاث اجوبة : اما أن العدد هو أكبر من الرقم المضمر أو أن العدد الذي زودته اقل من العدد المضمر أو أنه مساوٍ له وعند ذلك ستنتهي المحاولة وتعطيك نتيجة درجتك من 100 (يمكن أن تكون أكثر من مائة بالمائة) أما في ماعدا الحالة الأخيرة فعليك القيام بمحاولة جديدة حيث تعاد الكرة .
وبعد الانتهاء يسألك البرنامج فيما اذا كنت تريد محاولة اخرى وعند ذلك بالطبع سيختار رقماً آخر ولكن ضمن الحد المعطى في البداية وعليك أن تحاول أن تحزره . وبعد الانتهاء سيعطيك النتيجة من 100 ولكن ليس لمحاولة واحدة بل لجمل محاولاتك كلها منذ البداية وحتى تلك اللحظة .

```
10      INPUT "YOUR NAME";N$
20      D = 0
30      INPUT "MAXIMUM NUMBER"; A
40      I = 1
50      B = INT (A * RND (- TIME))
60      J = 1
70      INPUT "GUESS A NUMBER";C
80    IF C > B PRINT "TOO  BIG":J = J + 1 : GOTO  60
90  IF C<B PRINT "TOO  SMALL":J = J + 1 : GOTO  60
100 D = (LOG  (A) / LOG  (2) / J*100 + D*(I-1)) / I
110     PRINT "DEGREE =" D, N$
120 INPUT "D O YOU WANT T O TRY AGAIN Y / N";E$
130     IF E$ = "Y"THEN160
140     I = I + 1 : GOTO    40
160     END
```

البرنامج الثاني : قطة تصعد شجرة

البرنامج الآتي عبارة عن حزورة : تسألك الحاسبة عن الوقت الذي تستغرقه قطة لكي تصل أعلى شجرة .

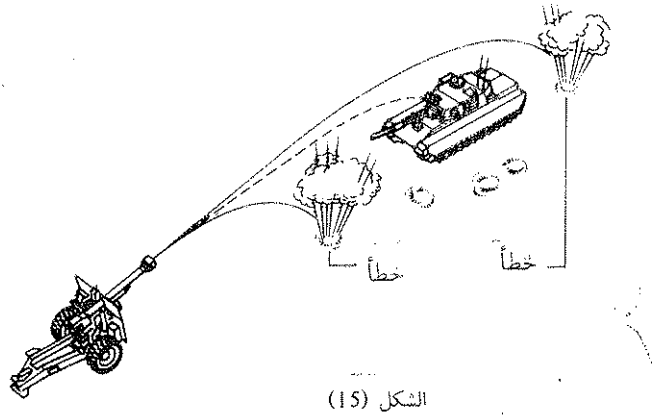
الحاسبة ستعطيك ارتفاع الشجرة وسرعة صعود القطة بالامتار لكل يوم . لكنها في نهاية كل يوم من صعودها تتعب فتزحف منحدرة الى الاسفل مقدار من الامتار ، ستخبرك به الحاسبة .

وسوف نترك لك فهم البرنامج بنفسك

```
10    H = INT (100 * RND (- TIME))
20    U = INT (H / 5 * RND (- TIME))
30    D = INT (U * RND (- TIME))
32    IF U = D THEN 20
33    IF U = 0 THEN 20
34    IF D = 0 THEN 20
35    H = (U-D) * INT ((H-U) / (U-D)) + U
50    PRINT "HIGHT OF TREE =" ;H ;"METERS"
60       PRINT "CAT SPEED =" ;U ;"METERS / DAY"
70    PRINT "CAT SLIDES DOWN" ;D ;"METERS /DAY"
80    INPUT "WHEN IT REACHES THE TOP"; A
90    IF (H-D) / (U-D) = A THEN 110
100    PRINT "WRONG.. TRY AGAIN". GOTO  80
110    INPUT "CORRECT .. DO YOU WANT TO TRY
                AGAIN ? Y / N" ;B$
120    IF B$ = "Y" THEN 10
```

البرنامج الثالث : اطلاق قذيفة على دبابة متحركة

لنفرض انك تريد اطلاق قذيفة على دبابة مهاجمة وانت لا تعلم بعدها عنك . كيف تتحكم بمدى القذيفة لكي تسقط على الدبابة؟ اذا اطلقت قذيفة فستجيبك الحاسبة بأنك قد عبرت المدى أو أن القذيفة كانت اقرب من المدى . وعند اصابتك الهدف تشير الى نجاحك . هذا مع العلم أن الدبابة ليست ثابتة في موقعها بل أنها تتقدم نحوك . لذلك فانك يمكن أن تخسر اذا ما وصلت الدبابة الى موقعك قبل أن تصيبها .



الشكل (15)

```
10  PRINT "TANK ATTACK !!"
20  PRINT "YOU ARE THE COMMANDER OF GUN
    BATTERY"
30  PRINT "AN ENEMY TANK IS ATTACKING"
40  PRINT "YOU ARE ORDERED TO DESTROY THE
    TANK BY FIRING YOUR GUN"
50  PRINT "YOU CAN ADJUST THE RANGE OF YOUR
    GUN"
60  PRINT "IF THE RANGE OF YOUR GUN IS
    CORRECT YOU WILL HIT THE TANK"
70  PRINT "EACH TIME YOU FIRE, THE TANK MOVES
    CLOSER TO YOUR GUN BATTERY"
```

```
80    PRINT "IF THE TANK REACHES THE BATTERY
      BEFORE BEING HIT YOU ARE OVERRUN AND
      MUST SURRENDER"
90    PRINT
100   D = 100
100   D = D - RND (20)
120   PRINT "WHAT RANGE (1 TO 100)?"
130   PRINT
140   INPUT R
150   IF D <= 20 THEN 230
160   IF ABS (R - D) <= 5 THEN 210
170   IF R < D PRINT "MISSED .. SHELL WAS
      SHORT .. TRY AGAIN"
180   PRINT
190   IF R >= D THEN PRINT "MISSED .. SHELL
      OVERSHOT .. TRY AGAIN"
195   PRINT
200   GO TO 110
210   PRINT "BOOM DIRECT HIT"
220   STOP
230   PRINT "GUN OVERRUN .. SURRENDER"
240   STOP
```

ملحق (١)
جدول أخطاء حاسبة الوركاء

* (Bad Subscript Error)? BS Error
خطأ في المتغيرات الموسومة (المصفوفات) حيث أن حجمها يزيد عن ما تم حجزه لها

* (Can't Continue Error)? CN Error
جملة الاستمرار CONT لا يمكن تنفيذها بعد حدوث خطأ في البرنامج أو بعد أن تم تعديل جزء من البرنامج

* (Duplicate Definition Error)? DD Error
تم استخدام متغير موسوم نفسه في جملتي بعد DIM أو أن جملة البعد DIM نفذت مرتين .

* ? Extra Ignored
عدد البيانات المعطاة بجملة DATA يزيد عن عدد المتغيرات المدخلة بواسطة جملة الادخال INPUT

* (Function Cell Error)? FC Error
خطأ في استدعاء دالة حيث أن العدد خارج الحيز المحدد

* (Illegal Direct Error)? ID Error
خطأ في استخدام جملة الادخال INPUT أو جملة تعريف الدالة DEFFN في النسق المباشر

* (Long String Error)? LS Error
طول سلسلة الحروف يزيد عن 255 حرفاً .

* (Missing Operand Error)? MO Error
هناك فقرة ضمن الجملة مفقودة : مثلاً فقدان الطرف الايمن من جملة احلال (اي بعد اشارة مساواة) .

* (Next without For Error)? NF Error
تم العثور على جملة NEXT دون المرور قبلها على جملة FOR

* (Out of Data Error)? OD Error
لم يتم العثور على بيانات عندما نفذت جملة READ

* (Out of Memory Error)? OM Error
البرنامج (طويل جداً . او أن هناك برنامج فرعي Subroutine يستدعي نفسه

(out of String Space Error)? OS Error
حجم المتغير الحرفي طويل جداً (أكثر من 255 حرفاً)

* (Over Flow Error)? OV Error
نتيجة الحسابات أما كبيرة جداً أو صغيرة جداً .

* (Return without Gosub! Error)? RG Error
تم المرور بجملة RETURN دون المرور قبلها على جملة GO SUB

* (Syntax Error)? SN Error
الجملة مكتوبة خطأ .

* (String Too Complex Error)? ST Error
الحسابات المتعلقة بسلسلة الحروف معقدة جداً .

* (Type Mismatch Error)? TM Error
تم استخدام الحروف والارقام المرتبطة معها بشكل خاطىء

* (Tape Read Error)? TR Error
خطأ في القراءة من الشريط .

* (Undefined Function Error)? NF Error
تم استخدام دالة تبتدىء بـ FN رغم عدم تعريفها مسبقاً بواسطة DEFFN

* (Undefined Line Number Error)? UL Error
ليس هناك سطر بهذا الرقم المشار اليه بواسطة جملة اقصد GOTO
أو جملة استدع GO SUB أو رتب RESTORE أو نفذ RUN . أو ربما كان هناك محاولة لحذف سطر لا وجود له اثناء عملية تصحيح

* (Division by Zero Error)? / O Error
محاولة القسمة على صفر

* (File Already open Error)? AO Error
أن السجل المطلوب فتحة كان مفتوحاً بالاصل أو أن رقم السجل المشار اليه مستخدم سابقاً أو أن هناك محاولة لتدمير سجل مفتوح .

* (Bad Allocation Table Error)?? AT Error
القرص لا يحتوي على فهرس السجلات ولا يمكن استخدامه . حاول أن تشكل القرص (FORMAT) ثم تستخدمه

* Bad File Number Error ??BN Error
رقم السجل المشار اليه يزيد عن الحد الاعلى المحدد خلال الابتداء .
أو أن رقم صفر قد اعطي لسجل متسلسل .

* (Bad File Mode Error)?? BM Error
هناك محاولة لمزج سجل بلغة وسيطة .
* (File Not Open Error)?? CF Error
السجل ذي الرقم المحدد غير مفتوح .
* (Disk Full Error)?? DF Error
ليس هناك حيز كافٍ على القرص
* (Bad Drive Number Error)?? DN Error
تمت الاشارة الى جهاز قرص غير مرتبط بالحاسبة .
* (Direct Statement in File Error)?? DS Error
محاولة لتحميل سجل بيانات (ليس سجل برامج)
* (Input Past End Error)?? EF Error
هناك محاولة لقراءة ما وراء نهاية سجل تسلسلي بدون استخدام جملة INPUT أو أنه في سجل عشوائي تمت الاشارة الى رقم قيد بواسطة جملة GET يزيد رقمه عن حجم السجل .
* (File Already Exists error)?? FE Error
السجل الجديد المسمى بواسطة جملة NAME موجود سابقاً على القرص
* (File Not Found Error)?? FF Error
فشل محاولة ايجاد سجل محدد
* (Field Overflow Error)?? FO Error
الطول الكلي لقيد واحد معرف بواسطة جملة FIELD يزيد عن 256 بايت

ملحق (2)
جدول اخطاء حاسبة صخر

0 Next Without FOR
لقد تم المرور بجملة NEXT في نهاية حلقة دون أن يسبقها المرور على جملة FOR التي هي بداية الحلقة

1 Syntax Error
خطأ في تركيب الجملة مثلاً فتح قوس دون غلقه او وجود خطأ في تهجئة ايعاز ما

2 RETURN without GOSUB
وجود جملة RETURN دون المرور مسبقاً على جملة GOSUB

4 Out of DATA
تم تنفيذ جملة READ وانتهت البيانات وتحتاج الجملة الى المزيد من البيانات

5 Illegal function call
تم المرور على متغير وسيط خارج المدى المقرر لدالة رياضية أو حرفية . ويمكن أن يحدث هذا الخطأ في حالات اخرى مثل :
أ) الرقم المميز لمتغير موسوم سالب أو كبير جداً بشكل غير معقول
ب) العدد المطلوب اخذ جذره التربيعي أو لوغاريتمه سالب .
جـ) استخدام دالة خاصة بالمستخدم USR دون تحديد عنوان البداية
د) استخدام جملة ERASE أو SWAP أو VARPTR دون تحديد المتغير
هـ) استخدام جمل معينة بشكل مخطوء مثل MID$ و LEFT$ و RIGHT$ و INP , OUT , WAIT , PEEK , POKE , TAB , SPC , STRING$, SPACE$, INSTR , GOTO , ON , GOSUB
و) أشارة الى رقم قيد سالب اثناء تنفيذ جملة PUT أو GET
ز) استخدام اوامر أو جمل رسوم في الشاشة 0 أو 1
ح) استخدام حروف غير مسموح بها بلغة GML أو MML
ط استخدام SPRITE في شاشة النسق 0

6 Overflow
نتيجة الحسابات كانت كبيرة جداً لا يمكن تبيانها بالاعداد المسموحة بلغة بيسيك . اما اذا حدث Underflow فيعني ذلك أن النتيجة صفر أو أن التنفيذ مستمر دون خطأ .

7 Out of memory

البرنامج طويل جداً أو فيه عدد كبير من حلقات FOR أو GOSUB أو متغيرات كثيرة .

8 Undefined line number

تمت الاشارة الى رقم جملة غير موجودة من خلال GOTO أو GOSUB أو IF .. THEN أو DELETE . ويمكن أن يحدث هذا الخطأ اذا كان هناك جملة تحتوي على رقم فقط ولاتحتوي على أي ايعاز .

9 Subscript out of range

تم الرجوع الى عنصر مصفوفة برقم خارج الحيز المحدد بجملة البعد DIM أو أن عدد الرموز المشيرة الى البعد خطأ (مصفوفة ذات بعد واحد اشير اليها بعنصر ذي بعدين مثلاً) .

10 Redimensioned array

اعطيت جملتان DIM للمصفوفة نفسها أو أن جملة DIM قد اعطيت بعد أن اعطت الحاسبة البعد الاعتيادي لها (عشرة عناصر) بسبب استخدام المصفوفة قبل جملة DIM

11 Division by zero

تمت محاولة للقسمة على صفر في تعبير ما أو أن المقدار صفر قد رفع الى اس سالب .

12 Illegal direct

جملة غير مسموح بها في النسق الاعتيادي ادخلت كايعاز

13 Type mismatch

تم اعطاء قيمة عددية لمتغير حرفي أو بالعكس أو أن الدالة التي تتوقع متغيراً عددياً قد اعطيت متغيراً حرفياً أو بالعكس .

14 Out of string space

المتغيرات الحرفية قد سببت لغة بيسيك لكي تستهلك الذاكرة الفارغة المخصصة والتي خصصت بواسطة جملة CLEAR

15 String too Long

هناك محاولة لاستحداث حروف طولها اكثر من 255 حرفاً

16 String formula too complex

تعبير حرفي طويل أو معقد . يجب تجزئة التعبير الى تعابير اقصر .

17 Can't Continue

هناك محاولة للاستمرار في برنامج :

أ) قد توقف بسبب خطأ

ب) قد تم اجراء تغيير عليه اثناء قطع Break
جـ) غير موجود اصلاً

18 Undefined User Function
تم استدعاء دالة معرفة من قبل المستخدم USR قبل أن يعرفها بواسطة جملة DEF

19 Device I /0 error
حدث خطأ في اشتغال جهاز ادخال أو اخراج

20 Verify error
محتويات الذاكرة ومحتويات السجل المسترجع مختلفتان . ربما يحدث هذا الخطأ عند استخدام جملة CLOAD?

21 No RESUME
ادخل روتين معالجة الاخطاء ولكنه لا يحتوي على جملة RESUME

22 RESUME without error
تم العثور على جملة RESUME قبل الدخول الى روتين معالجة الاخطاء

23 Unprintable error
ليس هناك رسالة خطأ محددة للظروف الموجودة

24 Missing Operand
هناك تعبير يحتوي على عامل مؤثر وليس هناك عامل مؤثر به . مثال ذلك حينما يتم استخدام جملة أو ايعاز يحتاج الى عوامل اجبارية لم يتم اعطاؤها

25 Line buffer overflow
تمت محاولة ادخال INPUT لسطر يحتوي على اكثر من 255 حرف

26 - 49 Unprintable error
ليس هناك رسالة خطأ محددة للظروف الموجودة .

50 Field overflow
جملة FIELD تحاول حجز مقدار من الذاكرة يزيد عن طول قيود سجل عشوائي

51 Internal Error
حدث عطب ببرامجيات الحاسبة الاساس للغة بيسيك . راجع الشركة مع اعطاء تفصيلات لظروف حصول العطب .

52 Bad file number
قام ايعاز أو جملة بالرجوع الى سجل برقم معين وكان ذلك السجل غير مفتوح من خلال جملة OPEN أو أن رقم السجل خارج المدى المحدد عند البدء .

53 File not found

إستخدمت ايعازات LOAD أو KILL أو NAME أو OPEN لسجلات غير موجودة على القرص الحالي .

54 File already open

سبق أن استخدمت جملة OPEN لسجل والآن استخدمتها مرة اخرى أو أن جملة KILL قد استخدمت لسجل سبق وأن فتح من خلال OPEN .

55 Input past end

تم طلب تنفيذ جملة INPUT بعد نفاذ البيانات . أو أن هناك محاولة للقراءة من سجل فارغ . لغرض تلافي هذا الخطأ استخدم دالة EOF للكشف عن نهاية السجل .

56 Bad file name

أستخدمت صيغة غير مسموح بها كاسم لسجل من خلال ايعازات LOAD أو SAVE أوKILLأو OPEN مثلاً عدد حروف اسم السجل اكثر من العدد المسموح .

57 Direct statement in file

تم العثور على جملة مباشرة اثناء تحميل LOAD لسجل نوع اسكي ASCII . لذلك فقد توقفت عملية التحميل .

58 Sequential I / O only

تم استخدام جملة GET أو PUT مع سجل سبق أن فتح بواسطة OPEN كسجل تسلسلي .

59 File not OPEN

تم استخدام جملة أو أمر ادخال أو اخراج مع سجل لم يسبق فتحه بواسطة جملة OPEN .

60 Bad FAT

جدول تنظيم السجلات (FAT) غير مرتب . ربما لم تتم عملية تشكيل FORMAT للقرص .

61 Bad file mode

تمت محاولة لاستخدام ايعاز PUT أو GET أو LOF مع سجل تسلسلي أو LOAD مع سجل عشوائي او محاولة تنفيذ جملة OPEN مع نسق سجل غير مناسب .

62 Bad drive name

تم استخدام اسم لمشغل drive غير مسموح به .

63 Bad sector number
يجب أن لا يحدث في الطراز الجديد للغة بيسيك الخاصة بالقرص .

64 File still open
لم يتم غلق السجل بواسطة ايعاز CLOSE بعد .

65 File already exist
السجل الذي تم تحديده بواسطة جملة NAME مطابق لاسم سجل موجود سابقاً على القرص .

66 Disk full
تم استخدام كل مجال الخزن على القرص .

67 Too many files
تمت محاولة لاستحداث سجل جديد (باستخدام SAVE او OPEN) مع أن كل فقرات الفهرس الـ 255 مملوءة .

68 Disk write protected
القرص يحوي على حماية ضد الكتابة عليه او لا يمكن الكتابة عليه .

69 Disk I/O error
حدث خطأ لا يمكن اصلاحه اثناء اشتغال جهاز ادخال أو اخراج .

70 Disk offline
مشغل القرص غير مربوط مباشرة .

71 Rename across disks
تمت محاولة اعادة تسمية سجل بواسطة مشغل جديد وهذا غير مسموح .

72 - 255 Unprintable error
ليس هناك رسالة خطأ محددة للظروف الموجودة .

ملحق (3)

معلومات عن حاسبة اكواريوس Aquarius مع جدول اخطائها

هذه الحاسبة من ارخص الحاسبات التي توفرت في الاسواق وتصنع في هونك كونك . ويحوي الطراز المطور منها على حروف عربية وامكانية للبرمجة باللغة العربية . وسنركز على الطراز البسيط في تبيان الفروق بينها وبين حاسبة صخر والوركاء . يبين الشكل (16) صورة لوحة المفاتيح لهذه الحاسبة .

وفيما يأتي بعض الملاحظات عن برمجة هذه الحاسبة :

1) مفتاح الارجاع مكتوب عليه RTN وهي مختصر RETURN . وهناك مفتاح للتصفير RST

2) يمكن بدل استخدام الايعاز PRINT استخدام علامة الاستفهام . مثلاً

```
10      PRINT 2 * 5
10      ?2 * 5
```

3) اذا انتهى السطر ولم تضغط على مفتاح الارجاع فإن البقعة المضيئة على الشاشة تبقى في آخر السطر مع اعطاء صوت اشارة الى أن عليك الضغط على مفتاح الارجاع .

4) المفتاح ← يقوم مقام مفتاح الحذف DEL الذي يحذف الحرف السابق

5) يمكن طبع رسالة على الشاشة بدون حتى استخدام كلمة PRINT أو علامة الاستفهام كالآتي :

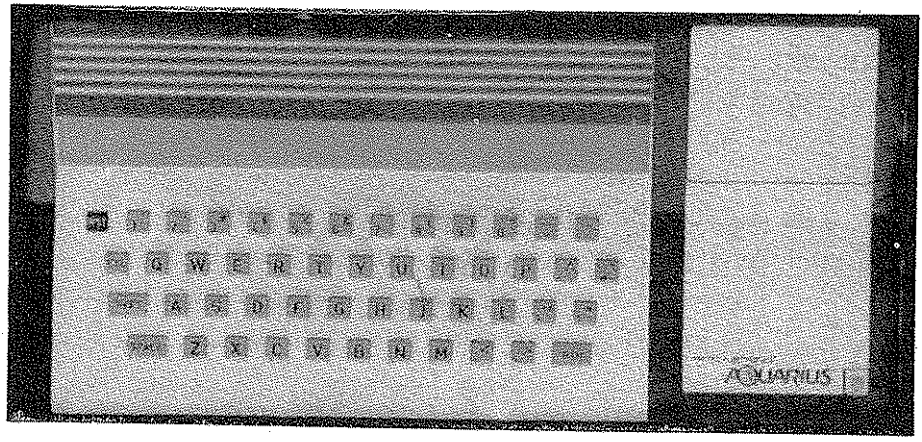
```
10      "WHAT IS YOUR NAME";
```

6) الدالة العشوائية تقبل دليلاً موجباً لها أو صفراً (بين قوسين) وفي الحالة الاخيرة فإن العدد العشوائي يعيد نفسه

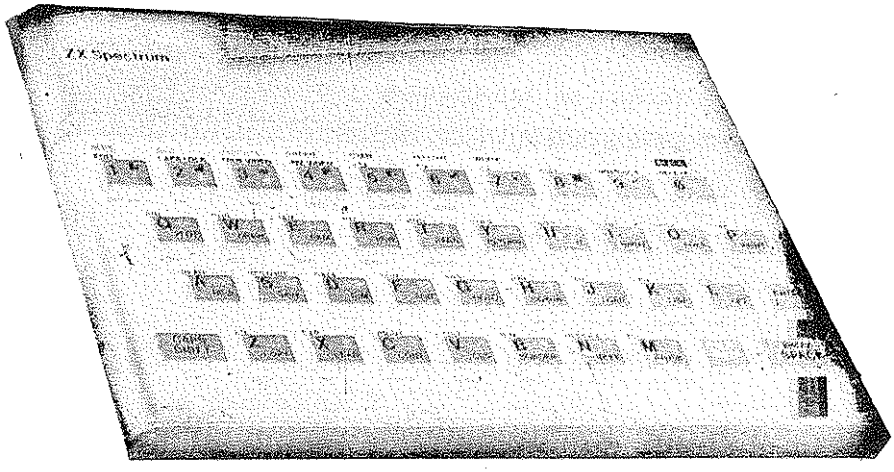
7) ليس هناك دالة للزمن TIME ولكن يمكن عمل ساعة توقيت بوسائل برمجية لا مجال لذكرها هنا .

جدول الاخطاء

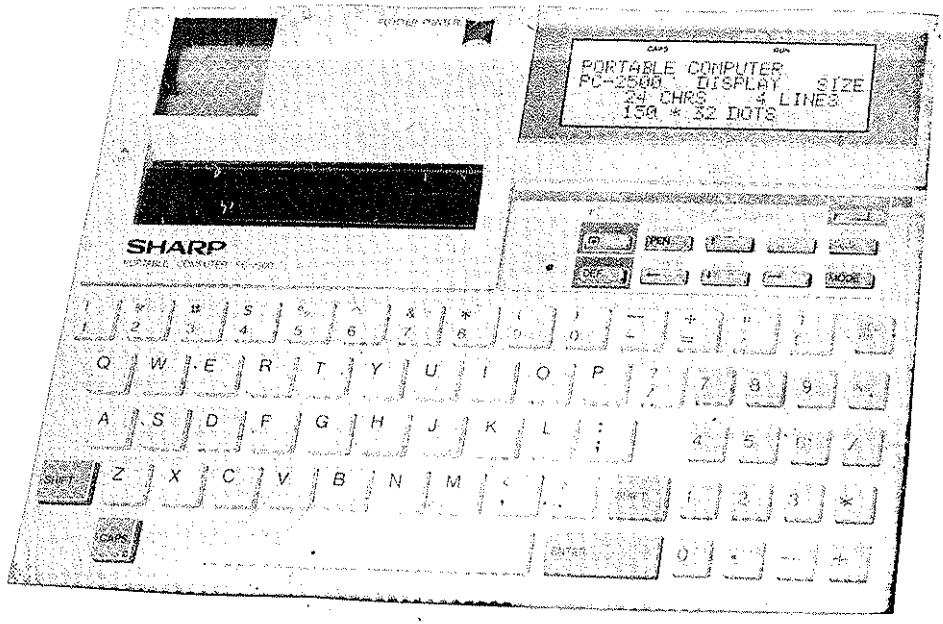
Bs	رسم المصفوفة خطأ لا يتفق مع جملة البعد
CN	لا يمكن الاستمرار بعد جملة STOP حيث ربما هناك خطأ اثناء الايقاف أو أن هناك برنامج في الذاكرة أو أن البرنامج قد توقف بسبب خطأ .
DD	بعد مكرر . أو أن الجملة الخاصة بالبعد DIM صودفت بعد أن اعطت الحاسبة بعداً للمصفوفة
/0	قسمة على صفر



الشكل (16) حاسبة اكواريوس



الشكل (17) حاسبة سنكلير – سبكترم



الشكل (18) حاسبة شارب

FC	استدعاء دالة بشكل خاطيء مثلاً محاولة ايجاد جذر لكمية سالبة أو بعد سالب لمصفوفة
ID	لا يمكن استخدام هذا الايعاز في النسق المباشر
LS	طول المتغير الحرفي اكثر من المسموح به (255)
MO	فقدان دليل الايعاز
NF	صودفت جملة NEXT بدون المرور على جملة FOR
OD	البيانات غير كافية حيث نفذت جملة READ وهي بحاجة الى بيانات اخرى
OM	نفاذ الذاكرة بسبب كبر حجم البرنامج أو حجم المصفوفة أو وجود متغيرات كثيرة أو وجود تعبير معقد جداً .
OS	عدم وجود ذاكرة كافية لخزن المتغير الحرفي الطويل جداً
OV	نتيجة العمليات الحسابية كبيرة جداً اكثر مما تسمح به لغة بيسيك
RG	وجود جملة ارجاع RETURN بدون جملة GOSUB
SN	خطأ نحوي بسبب خطأ في التهجئة أو حرف غير مسموح به
ST	قانون المتغير الحرفي معقد جداً
TM	الانواع غير متناسقة مثل اعطاء قيمة رقمية لمتغير حرفي
UF	دالة خاصة بالمستخدم غير معرفة سابقاً
UL	محاولة للتفرع الى سطر غير موجود بالذاكرة

ملحق (4)
معلومات عن حاسبة سنكلير سبكترم Sinclair Spectrum
مع جدول باخطائها

هذه الحاسبة من اول الحاسبات الرخيصة التي غزت الاسواق وخاصة في بريطانيا . وتمتاز بقابلياتها الواسعة وبأن الايعازات المختلفة عليها يتم ادخالها بشكل ايعاز كامل دفعة واحدة وليس حرفاً حرفاً . يبين الشكل (17) لوحة مفاتيح لهذه الحاسبة . ونظراً لان عدد الايعازات والرموز والاحرف اكثر بكثير من عدد المفاتيح فإن لكل مفتاح خمس وظائف أو اكثر أو اقل .

ويتم الانتقال من وظيفة الى اخرى أما بالتزحيف بواسطة مفتاح CAPSSHIFT أو بمفتاح SYMBOL SHIFT أو بوضع الالة في نسق مختلف . تشير البقعة على الشاشة الى نسق الاشتغال ويمكن أن تكون الانساق كالآتي :

الحرف K يشير الى النسق الخاص بالايعازات Keywords . وهذا النسق يظهر ذاتياً بعد النسق L اذا ما توقعت الحاسبة ايعازاً أو سطر برنامج

الحرف L للحروف . يحدث هذا النسق في كل الاحوال الاخرى اذا لم يتم استخدام أي من مفتاحي التزحيف

الحرف C للحروف الكبيرة باستخدام CAPS LOCK

الحرف E للنسق الاضافي للحصول على حروف اضافية ويتم ذلك بالضغط على مفتاحي التزحيف سوية

الحرف G للرسوم ويتم الوصول الى هذا النسق من خلال الضغط على مفتاح GRAPHICS (أي CAPS SHIFT والرقم 9) ويتم الخروج من هذا النسق بالطريقة نفسها

مثال : عند الضغط على المفتاح A فإن ذلك يعني :

الحرف a عند الضغط على المفتاح A لوحده

الحرف A عند الضغط على المفتاح A مع CAPS SHIFT

أيعاز READ بعد الضغط على مفتاحي التزحيف كليها ثم المفتاح A بشرط أن يكون رقم الجملة قد سبق ادخاله

الايعاز STOP بالضغط على مفتاحي الحروف الكبيرة CAPS SHIFT والتزحيف SYMBOL SHIFT

الايعاز NEW الايعاز الاعتيادي بعد ادخال رقم جملة (نسق الايعازات K).

الرمز ~ بالضغط على مفتاحي التزحيف سوية عند النسق الاضافي (E).

وينطبق كل ما تم شرحه عن البرمجة بلغة بيسيك على حاسبتي الوركاء وصخر في الفقرات (7-37) مع ملاحظة الاختلافات الآتية :

1) أن ايعاز LET ايعاز اجباري وليس اختياري . لذا يجب أن تبدأ جمل الاحلال بهذا الايعاز

2) أن جملة اذا تكون بصيغة

```
10    IF A = B THEN GOTO 100
```

حيث أن الايعاز GOTO في هذه الحالة اجباري ولايجوز حذفه

3) في جملة اذا ايضاً

```
20    IF A = B THEN LET C = 15
```

فإن ايعاز LET اجباري . أو بكلمة اخرى انه بعد كلمة THEN يجب أن يكون هناك ايعاز

4) يستخدم السهم للرفع الى اس بدل^

5) ايعاز CONTINUE كاملاً يقابل ايعاز CONT في حاسبتي صخر والوركاء .

6) ليس هناك امكانية لترقيم الجمل ذاتياً من قبل الحاسبة بل عليك ادخال الترقيم بنفسك وليس هناك امكانية لاعادة الترقيم . أي أن الايعازين AUTO RENUM غير مسموح بهما

7) اذا اردت انهاء البرنامج عليك استخدام ايعاز STOP وليس END لان الاخير لاوجود له

8) ايعاز اللوغاريتم الطبيعي هو LN هنا وليس LOG

9) ايعاز الدالة العشوائية هو RND ولكنه لايستخدم أي دليل (موجب أو سالب أو صفر) لكن يمكن تحديد نقطة بدء الدالة العشوائية بواسطة جملة اخرى هي (RANDOMIZE) يليها رقم بين 1 و 65535

جدول الاخطاء

0 OK
أكمل البرنامج بنجاح أو كان هناك قفز الى رقم سطر أكبر من أي سطر موجود .

1 NEXT without FOR
عدم وجود دليل الحلقة بسبب المرور على NEXT بدون مرور سابق على جملة FOR .

2 variable not found
يحدث عند استخدام متغير قبل تعريف قيمته بواسطة جملة احلال بسيطة LET أو قراءته بجملة READ أو جملة ادخال INPUT كما يحدث عند استخدام مصفوفة دون تعريف بعد لها DIM .

3 subscript wrong
دليل المتغير الموسوم اكبر من بعد المصفوفة أو أن هناك خطأ في دليل المتغير مثلاً أن يكون سالباً او اكبر من 65535 .

4 out of memory
عدم وجود محل في الذاكرة . لغرض المعالجة يمكن حذف الايعاز بواسطة DELETE ثم حذف جملة أو جملتين وبعد ذلك التحرك لحل المشكلة .

5 out of screen
جملة ادخال INPUT حاولت توليد اكثر من 23 سطر . وتحدث كذلك مع PRINT AT .

6 Number too big
الحسابات قد انتجت رقماً اكبر من 10^{38}

7 RETURN without GOSUB
صودفت جملة RETURN بدون المرور على GOSUB .

8 End of file
نهاية فايل عند التعامل مع جهاز قرص او شريط .

9 STOP Statement
بعد هذه جملة CONTINUE لا تعيد جملة STOP ولكن تستمر بالجملة التالية .

A Invalid argument
إن دليل هذه الدالة غير صحيح لسبب ما (تصادف مع SQR أو LN أو VSR وغيرها) .

B Integer out of range

عند تقريب عدد الى رقم صحيح كان الرقم خارج الحيز المناسب .

C Nonsense in BASIC

حروف متتالية لا تعطي مقدار ذي معنى .

D BREAK - CONT repeats

تم استخدام ايعاز BREAK اثناء استخدام احد الاجهزة الملحقة .
إن تصرف CONTINUE بعد ذلك يكون باعادة الجملة .

E Out of Data

حاولت القراءة بعد نفاذ البيانات .

F Invalid file name

تم استخدام ايعاز الحفظ SAVE لأسم فارغ او أكثر من 10 حروف .

G No room for line

ليس هناك مجال باقي في الذاكرة لاستيعاب ما تبقى من البرنامج .

H STOP in INPUT

بدأت بعض البيانات الخاصة بالادخال بجملة STOP . بعد هذا الخطأ يمكن استخدام CONTINUE ببساطة باعادة جملة الادخال .

I FOR without NEXT

بدأت حلقة بجملة FOR دون الوصول الى جملة NEXT .

J Invalid I/O device

خاص بالاجهزة الملحقة كجهاز القرص أو التسجيل .

K Invalid Colour

الرقم المحدد للون غير مناسب .

L BREAK into program

تم الضغط على مفتاح BREAK بين جملتين . رقم السطر المشار اليه هو للجملة قبل الايقاف . وعند ادخال CONTINUE ينتقل التنفيذ الى الجملة التالية دون أي تكرار .

M RAMTOP no good

العدد المحدد بواسطة RAMTOP اما كبير جداً او صغير جداً .

N Statement lost

هناك قفز الى جملة غير موجودة .

O Invalid stream

يصادف مع جهاز القرص او المسجلة .

P FN without DEF

استخدمت دالة دون تعريفها من قبل المستخدم .

Q Parameter error

خطأ في اعداد دليل الدالة أو أن احدهم من نوعية خاطئة .

R Tape loading error

التسجيل موجود على الشريط لكن لم تتم القراءة لسبب ما .

ملحق (5)

معلومات عن حاسبة شارب SHARP PC-2500 مع جدول باخطائها

هذه حاسبة متنقلة تختلف عن الحاسبات التي مر شرحها في أنها لا ترتبط بشاشة تلفزيونية بل شاشتها الصغيرة المشابهة لشاشات الساعات الالكترونية أو حاسبات الجيب وهي مكونة من بلورات سائلة . تستوعب هذه الشاشة اربعة اسطر كل سطر 24 حرفاً . ويرتبط بهذه الحاسبة جهاز طبع ورسم صغير يمكن بواسطته طبع النتائج على الورق او رسمها إن كانت بشكل منحنيات بيانية . انظر الشكل (18) ويمكن تشغيلها من مصدر التيار الكهربائي الذي يشحن بطارية قابلة للشحن داخل الحاسبة . ويمكن بعد ذلك استخدامها بمعزل عن مصدر التيار بالاعتماد على تلك البطارية لمدة قد تصل الى 15 ساعة اذا لم يستخدم جهاز الرسم مدة طويلة .

في هذه الحاسبة ثلاث امكانيات : لبرامج تجارية تدعى معالجة الجداول spread sheeats وبرنامج لخزن دليل الهاتف والثالث هو للبرمجة بلغة بيسيك . وسنركز على البرنامج الثالث :

أ) حول مفتاح التشغيل الى وضع ON .
ب) اختر لغة بيسيك بالضغط على المفتاح 3 .
جـ) ستجد أن الحاسبة تحولت الى نسق تنفيذ البرنامج RUN MODE .
د) اضغط على مفتاح النسق MODE حيث ستجد أن الحاسبة انتقلت الى نسق البرمجة PROGRAM MODE .
هـ) يمكنك الآن ادخال برنامجك بلغة بيسيك وبعد الانتهاء ارجع الى RUN MODE بالضغط على مفتاح MODE أيضاً ثم ادخل ايعاز التنفيذ مثل RUN .

إن الاختلافات الرئيسة بين اشتغال لغة بيسيك على هذه الحاسبة وما سبق شرحه على حاسبتي الوركاء وصخر يتلخص فيما يأتي : :

1) هناك ايعازان مختلفان للاخراج نظراً لوجود امكانية للاخراج على الشاشة واخرى على جهاز الطبع او الرسم . وهذان الايعازان هما PRINT للشاشة و LPRINT على جهاز الطبع .
2) هناك ايعازان مختلفان لعرض البرنامج اثناء التنفيذ هما LIST على الشاشة و LLIST على جهاز الطبع .
3) هناك دوال رياضية اضافية في هذه الحاسبة منها معكوس الجيب ASN (= حـا$^{-1}$) ومعكوس جيب التمام ACS ومعكوس الظل ATN

واللوغاريتمات الطبيعية LN واللوغارتيمات الاعتيادية LOG واستخدام قيمة النسبة الثابتة ط (PI) واستخدام الزاوية بالتقدير الستيني DEGREE ونصف القطري RADIAN والمئوي GRAD .

4) عند استخدام LIST أو LLIST تستخدم الفوارز لتحديد المدى مثلاً

10 LLIST 100 ,130

أي اعرض من السطر 100 الى 130 .

جدول الاخطاء

1 خطأ نحوي في تركيب المقادير مثل تعاقب اشارتي ضرب وقسمة أو خطأ في تهجئة بعض المتغيرات او نسيان الفوارز .

2 خطأ في الحسابات مثل القسمة على صفر أو استخدام عدد كبير جداً (10^{100} فاكثر) او ايجاد دالة غير موجودة مثل لوغاريتم عدد سالب .

3 دالة غير مسموح بها مثل استخدام مصفوفة بابعاد تزيد عن 255 أو رقم يعقب دالة يزيد عن الحد المسموح مثل WAIT 66000 .

4 استخدام رقم جملة اكثر من المسموح به (اكبر عدد 65279) أو استخدام رقم جملة غير موجود من خلال ايعازات GOTO أوGOSUBاو RUN أو LIST أو THEN .

5 ايعاز NEXT بدون FOR أو أن هناك اكثر من خمس مستويات للحلقات (احداها داخل الاخرى) أو أن هناك ايعاز اعادة RETURN بدون GOSUB أو ايعاز قراءة READ بدون بيانات DATA أو أن حيز البيانات قد نفذ .

6 فيض في الذاكرة فيما اذا اعطي بعد لمصفوفة كبيرة جداً أو أن البرنامج طويل جداً لا تستوعبه الذاكرة .

7 خطأ في الطبع باستخدام USING .

8 خطأ في اجهزة الادخال والاخراج وذلك عند ربط جهاز طبع خارجي أو مسجلة . ويعني أن هناك خطأ في الاتصالات . كما يظهر هذا الخطأ حينما تهبط الفولتية في البطارية الداخلية وتكون بحاجة الى شحن .

9 اخطاء اخرى لم تذكر سابقاً واكثر ما تحدث اذا حاولت استخدام بيانات حرفية بشكل A$ بينما كنت قد ادخلتها بشكل A .

هذا وإنه عند حدوث أي من هذه الاخطاء فإن الحاسبة تشير الى الجملة التي وقع فيها الخطأ وذلك فور الانتقال الى نسق البرمجة فيسهل تصحيح الخطأ .

ملحق (6)
جدول مقارنة الايعازات بين الحاسبات الخمسة

الحاسبة / الايعاز	صخر	اتوركاء	اكواريوس	سنكلير	شارب
ABS	√	√	√	√	√
ATN	√	√	√	√	√
AUTO	√	×	√	×	×
CONT	√	√	√	CONTINUE ولكن بصيغة	√
COS	√	√	√	√	√
DATA	√	√	√	√	√
END	√	√	√	×	√
EXP	√	√	√	√	√
FOR/NEXT	√	√	√	√	√
GOTO	√	√	√	√	√
IF/THEN	√ وفيها امكانية ELSE ايضاً	√	√	√ يجب أن يعقبه ايعاز	√
INPUT	√	√	√	√	√
INT	√	√	√	√	√
LET	√ اختيارية	√ اختيارية	√ اختيارية	√ اجبارية	√ اختيارية
LIST	√	√	√	√	√ ومتوفر معها LLIST
LOG	√	√	√	√ ولكن بصيغة LN	√ متوفر كذلك LN
NEW	√		√	√	√
PRINT	√	√ يمكن استخدام؟	√ يمكن استخدام؟	√ يمكن استخدام؟	√ ومتوفر معها LPRINT
RND	√	√	√ تقبل فقط دليل موجب وصفر	√ يجب أن يسبقها Randomize	√
READ	√	√	√	√	√
REM	√	√	√	√	√
RENUM	√	√	√	×	×
RUN	√	√	√	√	√
SIN	√	√	√	√	√
SQR	√	√	√	√	√
STOP	√	√	√	√	×
TAN	√	√	√	√	√
TIME	√	√	×	×	×

√ متوفر ×غير متوفر

ملحق (7)
اجوبة التمارين

```
11-1 1.5
11-2 – 4
11-3 4096
15-1 10  M = 15
     20  PRINT M,"MILES =", 1.609 * M, "KILOMETERS"
15-2 10  H = 7
     20  R = 3
     30  PRINT "VOLUME OF CONE=" 3.14159*R ^2 * H /3
15-3 10  S = 120
     20  T = 0.1 * 5
     30  P = S-T-15
     40  PRINT "SALLARY =", S, "TAX =", T,
         "FIXED COSTS =", 15, "NET INCOME =", P
20-1 10  I = 7
     20  PRINT I
     30  I = I + 7
     40  GOTO  20
20-2 10  I = 1904
     20  PRINT I
     30  I = I + 4
     40  GOTO  20
20-3 10  I = 1
     20  PRINT I, "INCHES =", 2.54 * I, "CMS"
     30  I = I + 1
     40  GOTO  20
22-2     استبدل اسماء الايام في الجمل 20-80 بحيث تبدأ بيوم السبت
         واستبدل رقم الجملة 45 بالرقم 85 وعدد البيانات ثلاثة فقط
23.1 10  READ P, C
     20  READ Y
```

```
    30  P = P * C
    40  PRINT "POPULATION =", P, "AT YEAR", Y + 1
    50  GOTO    20
    60  DATA 12 000 000, 0.03
    70  DATA 75, 76, 77, 78, 79, 80, 81, 82, 83, 84
23.2 10  READ A, B, C
    20  PRINT "SALARY =", 10 * A + B + 0.05 * C
    30  GOTO    10
    40  DATA 10,5,6, 12,4,8, 10,9,2, 15,7,2,
        9,8,6
23.3 10  READ V,A
    20  READ T
    30  PRINT "VELOCITY AT", T, "SECONDS =",
        V + A * T
    40  GOTO    20
    50  DATA 10, 0.1, 0, 5, 10, 15, 20
24.1    لتحوير برنامج مثال 22.3 لكي يتوقف بدون توقف اضطراري
        يمكن استبدال جملة 40 بالجملة
    41  IF I < 17 THEN 20
28.1 10  FOR I = 1 TO 20
    20  PRINT I, "INCHES =" 2.54 * I, "CMS"
    30  NEXT I
28.2 71  IF H3 > 0 THEN 80
    72  INPUT "IS FIRST TIME THE DAY
        BEFORE?"; C$
    73  IF C$ = "YES" THEN 79
    74  PRINT "YOU MADE A MISTAKE IN
        GIVING SECOND TIME BEFORE FIRST"
    75  GOTO    80
    79  H3 = H3 + 24

28.3 10   INPUT A,D
     20   IF D > = 1 THEN 50
```

```
        30   PRINT "THE CUBE WILL SINK"
        40   GOTO   60
        50   PRINT "THE CUBE WILL FLOAT.
             HIGHT OF CUBE ABOVE WATER LEVEL
             = ",(D-1) * A
        60   STOP
32.1    10   FOR I = 1 TO 10 باستخدام ثلاث حلقات متتالية
        20   PRINT I ,SQR (I)
        30   NEXT
        40   FOR I = 10 TO 100 STEP 10
        50   PRINT I ,SQR (I)
        60   NEXT I
        70   FOR I = 100 TO 1000 STEP 100
        80   PRINT I ,SQR (I)
        90   NEXT I

                                  ويمكن باستخدام حلقتين متداخلتين
        10   FOR K = 1 TO 3
        20   FOR I = 1 TO 10
        30   N = I * 10 ^(K-1)
        40   PRINT N ,SQR (N)
        50   NEXT I : NEXT K
32.2    10   S = 0 : N = 100
        20   FOR I=1 TO 10
        30   PRINT N : S = N + S = : N = N * 1.1
        40   NEXT I
        50   PRINT S

32.3    10   INPUT M
        20   FOR I = 1 TO M
        30   READ N,B
        40   NEXT I
        50   FOR I = 1 TO N
        60   A = ( I + B ) - 7 * INT (( I + B) / 7)
```

```
70    IF A = 1 THEN C$ = "SATURDAY"
80    IF A = 2 THEN C$ = "SUNDAY"
90    IF A = 3 THEN C$ = "MONDAY"
100   IF A = 4 THEN C$ = "TUESDAY"
110   IF A = 5 THEN C$ = "WEDENSDAY"
120   IF A = 6 THEN C$ = "THURSDAY"
130   IF A = 0 THEN C$ = "FRIDAY"
140   PRINT I ,C$
150   NEXT I
160   DATA 31, 4, 28, 7, 31, 7, 30, 3, 31, 5,
      30, 1, 31, 3, 31, 6, 30, 2, 31, 4, 30, 7,
      31, 2
```

محتويات الجزء التالي

سيحوي الجزء التالي من هذا الكتاب إن شاء الله تعالى على مواضيع عديدة منها ما يأتي :

* المزيد عن تشغيل حاسبات الوركاء وصخر واكواريوس وسنكلير وشارب .
* المتغيرات الموسومة والايعازات الخاصة بها .
* الروتينات الفرعية وايعازاتها .
* استخدام الرسوم على الشاشة وكيفية برمجتها .
* استخدام الايعازات الموسيقية على الحاسبة الالكترونية .
* المزيد من الدوال المكتبية .
* أمثلة عديدة وتمارين متنوعة مع حلولها .
* برامج تسلية جديدة أكثر تعقيداً وأثارة مما في هذا الجزء .

وآخر دعوانا ان الحمد لله رب العالمين

د . محمد زكي محمد خضر
بواسطة ص . ب ١٣
الموصل ـ العراق

المصادر

1. فورتران ـ مدخل الى الحاسبات الالكترونية
د . محمد زكي محمد خضر و نبيل خليل عمر ـ جامعة الموصل ١٩٨٣

2. Peter Lafferty, Introduction to computing Simon & Schusters, Inc. 1983
3. Spenser, A Guide to Basic Programming, 2nd Ed. Addison-Wesley , 1975.
4. EIC, Personal computer, PC-6001 Bmk II N60m Basic Manual
5. EIC, Personal Computer, PC-6001 Bmk II Users Manual
6. Yamaha, MSX Basic Reference Manual, 1984
7- Sinclair, ZX Spectrum, Basic Programming Sinclair Research Ltd. 3rd Ed. 1983
8. ECICO Electronics Ltd. Aquarius Manual
9. Sharp Portable Computer PC-2500 Operation Manual
10. Personal Computer world , wall chart Basic Conversion Chart

11. شركة الصناعات الالكترونية :
مختصر ايعازات لغة بيسك العربي على الحاسبة الشخصية الوركاء 6001

المحتويـــات

الفقرة	الموضـــوع	الصفحة
	بين يدي الكتاب	5
1	هذا الكتاب	7
2	ما هي الحاسبة الالكترونية	7
3	حاسبة الوركاء	8
4	تشغيل حاسبة الوركاء	11
5	وظائف المفاتيح على حاسبة الوركاء	11
6	حاسبة صخر	13
7	الخطوة الاولى في البرمجة	15
8	الجمل التنفيذية والجمل البرمجية	16
9	هيأة جملة البرمجة	17
10	استخدام الحاسبة الالكترونية لاجراء العمليات الحسابية البسيطة	20
11	تمارين	22
12	الشرح والحساب معاً	22
13	جملة الاحلال الحسابية	23
14	أمثلة	26
15	تمارين	28
16	ايعاز الادخال	28
17	المتغيرات الحرفية	30
18	جملة تنفيذية اخرى : أعرض او سطِّر	32
19	تغيير تسلسل تنفيذ البرنامج : جملة اقصد	33
20	تمارين	35
21	جملة القراءة والبيانات	35
22	أمثلة	39
23	تمارين	44
24	الانتقال الشرطي : اذا .. اذن	45
25	مثال على جملة اذا .. اذن	48
26	ما هي المخططات الانسيابية	52
27	أمثلة على المخططات الانسيابية	54
28	تمارين	62

29 الحلقات : جملة من .. كرر 63
30 امثلة على الحلقات 65
31 تداخل الحلقات 72
32 تمارين 75
33 ايعازات تنفيذية اضافية 76
34 ايعازات برمجية اضافية 78
35 صيغ برمجية اخرى 78
36 الدوال المكتبية 80
37 برامج تسلية 83
ملحق 1 ـ جدول اخطاء حاسبة الوركاء 87
ملحق 2 ـ جدول اخطاء حاسبة صخر 90
ملحق 3 ـ معلومات عن حاسبة اكواريوس مع جدول اخطائها 95
ملحق 4 ـ معلومات عن حاسبة سنكلير سبكترم مع جدول اخطائها 98
ملحق 5 ـ معلومات عن حاسبة شارب مع جدول اخطائها 103
ملحق 6 ـ جدول مقارنة الايعازات بين الحاسبات الخمسة 105
ملحق 7 ـ اجوبة التمارين 106
محتويات الجزء التالي 109
المصادر 110
المحتويات 111