

المقدمة

قبل التحدث عن مبادئ البرمجة بلغة بييسك أود أن أشير إلى أن منهج وزارة المعارف يحتوي على مقدمة عن أنواع البرامج و البرمجيات و لغات البرمجة، لن نجدها مذكورة هنا في صفحات هذا الموقع. و يمكنك الرجوع لكتاب المقرر.

إن جميع البرامج تقوم بعمليات أساسية هي:

١. الحصول على البيانات المدخلات و ذلك بقراءتها من لوحة المفاتيح مثلاً أو تضمين البيانات داخل البرنامج، كما سنتعلم ذلك لاحقاً.
٢. المعالجة و هي التعامل مع البيانات التي تم الحصول عليها في مرحلة المدخلات و ذلك بتنفيذ العمليات الحسابية مثلاً على هذه البيانات.
٣. المخرجات و هي إخراج المعلومات أو النتائج مطبوعة على الطابعة أو على شاشة الحاسب و هذه المرحلة الأخيرة هي الهدف من كتابة البرنامج.

المدخلات ← المعالجة ← المخرجات

و قبل البدء بكتابة البرنامج لابد من النظر على أنه مسألة يجب تحليلها و التفكير فيها لمعرفة طريقة حلها، ثم ترجمة فكرة الحل إلى برنامج بلغة بييسك. و يمكن القول بأن هناك خطوات لحل البرنامج هي:

- تعريف المسألة.
- تحديد خطوات الحل (الخوارزمية).
- ترجمة الخوارزمية إلى برنامج بلغة بييسك.
- تنفيذ البرنامج و تصحيح الأخطاء إن وجدت.

و سيأتي شرح هذه الخطوات بالتفصيل في الدروس القادمة.

الخوارزميات

هي وسيلة لحل المسائل الحسابية ابتكرت من قبل العالم المسلم الخوارزمي. و تقوم فكرة الخوارزمية على تحليل المسألة التي بين يديك لتصل إلى حلها، و من ثم كتابة و حصر هذه الأفكار حتى يتسنى لأي شخص يقرأ هذه الخوارزمية بفهم الحل مباشرة. و الخوارزمية تكتب باللغة العربية البسيطة بغرض أن تشرح فيها حل المسألة و لكن باختصار.

إن مايجب أن تذكره في الخوارزمية هي المدخلات التي سيقراها البرنامج و كذلك العمليات التي سيقوم بها البرنامج سواء كانت عمليات حسابية أو غيرها. ثم تحدد المخرجات التي سيخرجها برنامجك. إذا ما حددت هذه الأساسيات سيتمكن أي مبرمج من كتابة البرنامج اعتماداً على الخوارزمية التي كتبها و بأي لغة يريد.

مثال: أكتب خوارزمية برنامج يقوم بطباعة مجموع طالب في مادة الحاسب. فما هي الخوارزمية اللازمة لحل هذا البرنامج.

الحل:

- قراءة درجة أعمال السنة (س).
- قراءة درجة الاختبار الفصلي (ص).
- احسب مجموع درجات الطالب $م = س + ص$.
- اطبع الدرجة النهائية (م).

المخطط الانسيابي

هي وسيلة مفيدة للتعبير عن فكرة حل المسألة برسم مخطط انسيابي لطريقة الحل. و تستخدم أشكال خاصة في الرسم للتعبير عن عملية معينة من عمليات البرمجة. و المخطط يبدأ من أعلى الصفحة ثم ينساب و ينزل للأسفل، و الجدول التالي يبين هذه الأشكال المستخدمة للتعبير عن عمليات البرمجة:

الرمز	الغرض	توضيح
	بداية / نهاية	يدل هذا الشكل على بداية المخطط إذا وضع في أعلى الصفحة أما في النهاية فهذا معناه أن الرسم قد انتهى
	إدخال / إخراج	الشكل التالي يدل على وجود عملية يتم إجراؤها وهي قراءة بيانات أو إخراج بيانات
	عملية	يدل هذا الشكل على وجود عملية حسابية
	قرار	يدل على وجود عملية مقارنة بين قيمتين بحيث نتكأ قرار معين بناءً على هذا القرار يتم تحديد الاتجاه الذي سوف يسير فيه المخطط الانسيابي
	خط انسيابي	يبين هذا الشكل الاتجاه الذي يسير فيه البرنامج

و للرسوم الانسيابية أنواع هي:

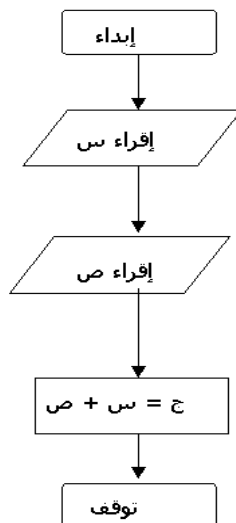
١. المخطط الانسيابي متتابعي.
٢. المخطط الانسيابي المتفرع.
٣. المخطط الانسيابي ذو الدورانات المتكررة.

إن اختيار أي نوع يعتمد على نوع المسألة التي نقوم بحلها. و لن نتضح طريقة استخدام هذه الأشكال في بناء المخطط الانسيابي إلا بعد عرض عدد من الأمثلة.

المخطط الانسيابي متتابعي

إن أبسط أنواع البرامج هو البرنامج التي تأتي أفكاره و خطوات حله متسلسلة وراء بعضها البعض، خطوة بعد خطوة حيث أنك عندما تقرأها تحس أنها فقرة واحدة متصلة. و لتمثيل هذا النوع من البرامج فإننا نرسم مخطط انسيابي متتالي لا يوجد به أي تفرعات. و لكي نتضح الفكرة أكثر و أكثر فسوف نقوم برسم المخطط الانسيابي لبرنامج يقرأ عدد ين ثم يقوم بطباعة ناتج الجمع، قبل رسم المخطط الانسيابي أقترح أن نقوم بكتابة الخوارزمية كنوع من التمرين.

الخوارزمية:



- إبداء.
- اقرأ العدد الأول (س).
- اقرأ العدد الثاني (ص).
- اجمع العدد (س) و (ص).
- اطبع المجموع.

المخطط الانسيابي:

الشرح: كما تلاحظ في الرسم السابق أن المخطط بدأ برمز البداية ثم بعد ذلك وضع المخطط أن هناك بعد البداية عملية قراءة للرقم (س) و هو الرقم الأول ثم تلاها قراءة للرقم الثاني (ص) ثم قام المخطط بتنفيذ عملية حسابية و هذا ما يدل على رمز المستطيل الذي يعني عملية حسابية. و في النهاية بعد ذلك سيتوقف تنفيذ البرنامج كما دلت علامة "توقف".

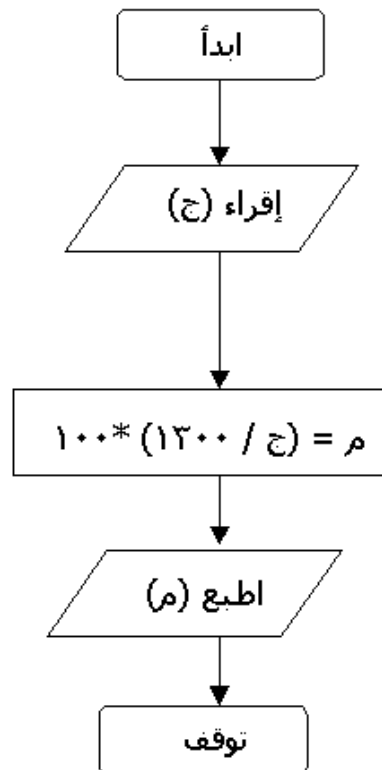
و حتى نتضح الفكرة أكثر إليك مزيد من الأمثلة.

مثال: اكتب خوارزمية برنامج يقوم بقراءة مجموع درجات الطالب ثم يوجد نسبته المئوية إذا علمت أن الدرجة النهائية هي ١٣٠٠ درجة، و ارسم المخطط الانسيابي لهذا البرنامج.

الخوارزمية

- ابدأ.
- إقرأ مجموع درجات الطالب (ج).
- احسب النسبة المئوية (م) حيث $م = (ج / ١٣٠٠) * ١٠٠$
- اطبع (م)
- توقف.

المخطط الانسيابي (تتابعي):



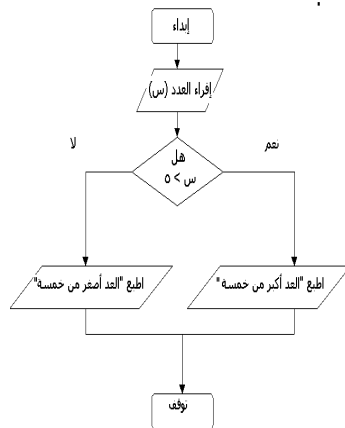
المخطط الانسيابي المتفرع

هناك عدد من البرامج فيها درجة بسيطة من التعقيد، و ذلك لاحتواء البرنامج على عدد من الشروط التي تستوجب عليك تنفيذ أمر معين إذا كان الشرط صحيحاً. أما إذا كان الشرط خاطئاً فإن البرنامج سيتفرع لينفذ أمر آخر. و من هذه البرامج ما تقوم بتحديد إذا كان الطالب ناجح أم لا، أو بتحديد إذا كان الرقم سالب أم موجب. و سيتضح كيفية المخطط الانسيابي المتفرع من المثال التالي حيث يقوم البرنامج بقراءة عدد ثم يطبع رسالة تبين هل العدد أكبر من خمسة أم أصغر من خمسة.

الخوارزمية:

- إبدأ
- إقرأ العدد (س).

- هل العدد (س) أكبر من خمسة.
- إذا كان العدد أكبر من خمسة اطبع "العدد أكبر من خمسة".
- إذا كان العدد أصغر من خمسة اطبع "العدد أصغر من خمسة".
- توقف.



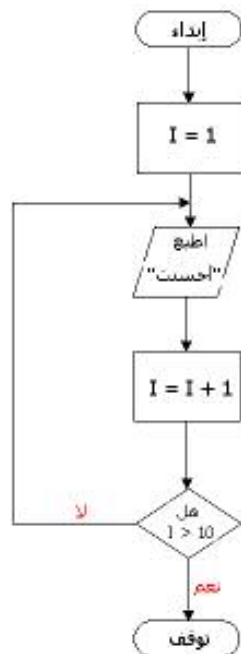
المخطط الانسيابي

شرح المخطط:

لاحظ من المخطط الانسيابي السابق أن المخطط يبدأ بقراءة العدد (س) ثم بعد ذلك قام بمقارنة العدد المقروء بالعدد خمسة لنرى إذا ما كان العدد المقروء أكبر من العدد خمسة أو أصغر من العدد خمسة. وبناءً على نتيجة هذه المقارنة سيتفرع المخطط الانسيابي إلى اليمين إذا كان الشرط صحيح (نعم) أو إلى اليسار إذا كان الشرط خاطئ (لا). فلو كان العدد المقروء مثلاً هو ٢٥ فإن البرنامج سيطلع جملة "العدد أكبر من خمسة" أما إذا كان العدد ٣ أي أصغر من خمسة فيتم طباعة جملة "اطبع العدد أصغر من خمسة".

المخطط الانسيابي ذو الدورات المتكررة.

قد تخيفك هذه التسمية في البداية ولكنها تدل على شئ بسيط و واضح. حيث أن هناك عدد من البرامج التي تكون في أحد مراحل تنفيذ تستلزم أن يقوم البرنامج بتكرار جزء معين من عمليات البرنامج أكثر من مرة. وبالطبع سيتضح المغزى من هذا النوع من المخططات مع المثال التالي الذي يقوم بطباعة عبارة "أحسننت" عشر مرات.



فلو تأملت المثال السابق وجدت أنه في البداية قام بوضع المتغير I بقيمة واحد ليستغله في عملية عدد مرات طباعة كلمة "أحسننت" ففي المرة الأولى يكون قيمة I هي واحد ثم بعد ذلك يقوم المخطط بطباعة عبارة "أحسننت". بعد ذلك نزيد قيمة المتغير I بوحد لتصبح (٢) وبذلك نعرف أننا سنقوم بالطباعة للمرة الثانية. بعد ذلك نختبر قيمة I هل هي أكبر من ١٠ أي هل وصل عدد طباعة عبارة "أحسننت" عشر مرات أم لا. فإذا كانت الإجابة بلا فإن التفرع يعود ليكرر عملية الطباعة وزيادة قيمة العدد I بوحد. وهكذا تكرر عملية الطباعة طالما أن عدد التكرارات لم يصل إلى ١٠ مرات. فإذا تجاوزت قيمة المتغير I القيمة ١٠ فهذا يعني أن الشرط أصبحت قيمته بنعم وبذلك يتوقف البرنامج عن العمل.

أسئلة و تمارين

سيكون التركيز في المسائل على رسم المخططات الانسيابية كون أغلب الأسئلة تُركز عليها.

ارسم مخططاً انسيابياً يقوم بطباعة الأعداد من ١ - ١٠.

الحل...

المخطط الانسيابي تتابعي

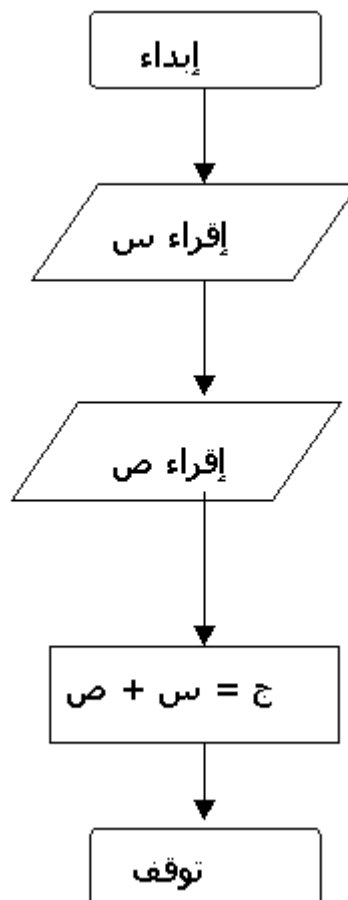
إن أبسط أنواع البرامج هو البرنامج التي تأتي أفكاره و خطوات حله متسلسلة وراء بعضها البعض، خطوة بعد خطوة

حيث أنك عندما تقرأها تحس أنها فطرة واحدة متصلة. ولتمثيل هذا النوع من البرامج فإننا نرسم مخطط انسيابي متتالي لا يوجد به أي تفرعات. ولكي نتضح الفكرة أكثر وأكثر فسوف نقوم برسم المخطط الانسيابي لبرنامج يقرأ عدد ين ثم يقوم بطباعة ناتج الجمع، قبل رسم المخطط الانسيابي أقترح أن نقوم بكتابة الخوارزمية كنوع من التمرين.

الخوارزمية:

- إبداء.
- اقرأ العدد الأول (س).
- اقرأ العدد الثاني (ص).
- اجمع العدد (س) و (ص).
- اطبع المجموع.

المخطط الانسيابي:



الشرح: كما تلاحظ في الرسم السابق أن المخطط بدأ برمز البداية ثم بعد ذلك وضع المخطط أن هناك بعد البداية عملية قراءة للرقم (س) و هو الرقم الأول ثم تلاها قراءة للرقم الثاني (ص) ثم قام المخطط بتنفيذ عملية حسابية و هذا ما يدل على رمز المستطيل الذي يعني عملية حسابية. وفي النهاية بعد ذلك سيتوقف تنفيذ البرنامج كما دلت علامة "توقف".

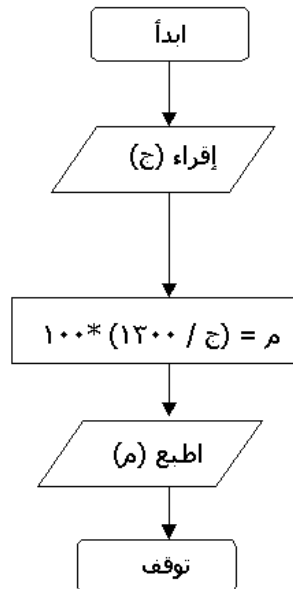
و حتى نتضح الفكرة أكثر إليك مزيد من الأمثلة.

مثال: اكتب خوارزمية برنامج يقوم بقراءة مجموع درجات الطالب ثم يوجد نسبته المئوية إذا علمت أن الدرجة النهائية هي ٣٠٠ درجة، و ارسم المخطط الانسيابي لهذا البرنامج.

الخوارزمية

- ابدأ.
- إقرأ مجموع درجات الطالب (ج).
- احسب النسبة المئوية (م) حيث $م = (ج / ١٣٠٠) * ١٠٠$
- اطبع (م)
- توقف.

المخطط الانسيابي (تتابعي):



ارسم مخططاً انسيابياً يقوم بطباعة الأعداد الزوجية من ١ - ١٠.

الحل...

المخطط الانسيابي المتفرع

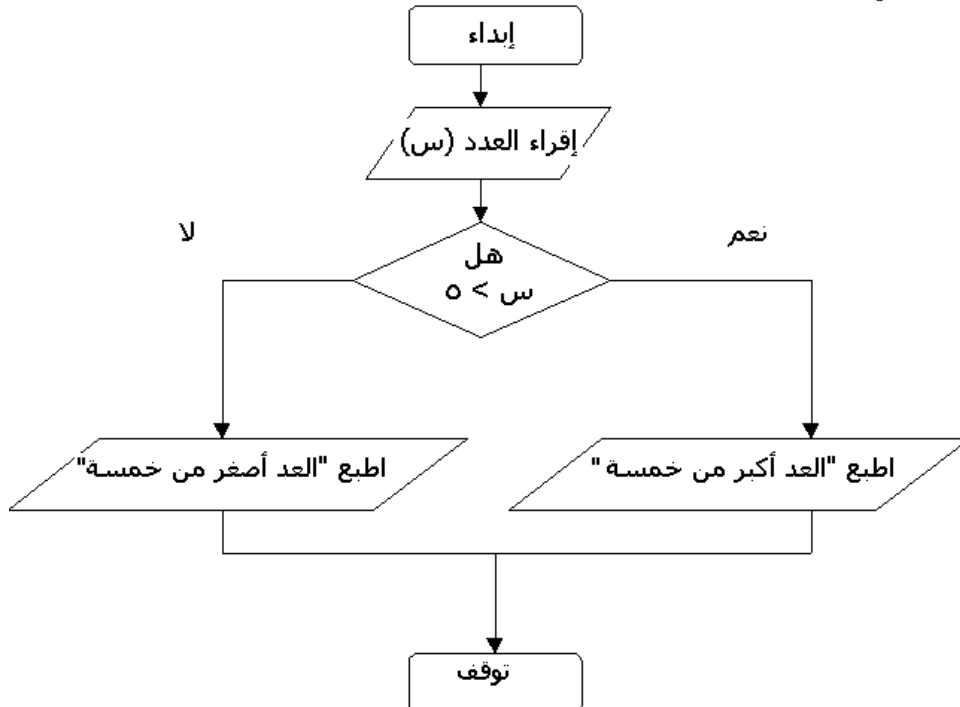
هناك عدد من البرامج فيها درجة بسيطة من التعقيد، وذلك لاحتواء البرنامج على عدد من الشروط التي تستوجب عليك تنفيذ أمر معين إذا كان الشرط صحيحاً. أما إذا كان الشرط خاطئاً فإن البرنامج سينتقل لينفذ أمر آخر. ومن هذه البرامج ما تقوم بتحدد إذا كان الطالب ناجح أم لا، أو بتحديد إذا كان الرقم سالب أم موجب. وسيوضح كيفية المخطط الانسيابي المتفرع من المثال التالي حيث يقوم البرنامج بقراءة عدد ثم يطبع رسالة تبين هل العدد أكبر من خمسة أم أصغر من خمسة.

الخوارزمية:

- إبداء
- إقرأ العدد (س).
- هل العدد (س) أكبر من خمسة.
- إذا كان العدد أكبر من خمسة اطبع "العدد أكبر من خمسة".

- إذا كان العدد أصغر من خمسة اطبع "العدد أصغر من خمسة".
- توقف.

المخطط الانسيابي



شرح المخطط:

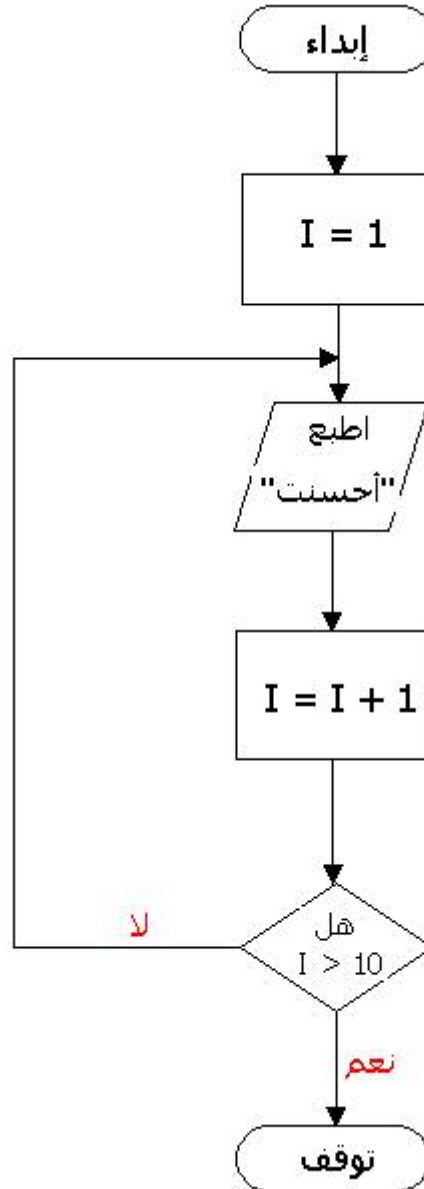
لاحظ من المخطط الانسيابي السابق أن المخطط يبدأ بقراءة العدد (س) ثم بعد ذلك قام بمقارنة العدد المقروء بالعدد خمسة لنرى إذا ما كان العدد المقروء أكبر من العدد خمسة أو أصغر من العدد خمسة. وبناءً على نتيجة هذه المقارنة سيتفرع المخطط الانسيابي إلى اليمين إذا كان الشرط صحيح (نعم) أو إلى اليسار إذا كان الشرط خاطئ (لا). فلو كان العدد المقروء مثلاً هو ٢٥ فإن البرنامج سيطبع جملة "العدد أكبر من خمسة" أما إذا كان العدد ٣ أي أصغر من خمسة فيتم طباعة جملة "اطبع العدد أصغر من خمسة".

ارسم مخططاً انسيابياً يقوم بطباعة الأعداد الفردية من ١ - ١٠.

الحل...

المخطط الانسيابي ذو الدورانات المتكررة.

قد تخيفك هذه التسمية في البداية و لكنها تدل على شئ بسيط و واضح. حيث أن هناك عدد من البرامج التي تكون في أحد مراحل تنفيذ تستلزم أن يقوم البرنامج بتكرار جزء معين من عمليات البرنامج أكثر من مرة. و بالطبع سيتضح المغزى من هذا النوع من المخططات مع المثال التالي الذي يقوم بطباعة عبارة "أحسننت" عشر مرات.



فلو تأملت المثال السابق وجدت أنه في البداية قام بوضع المتغير I بقيمة واحد ليستغله في عملية عدد مرات طباعة كلمة "أحسننت" ففي المرة الأولى يكون قيمة I هي واحد ثم بعد ذلك يقوم المخطط بطباعة عبارة "أحسننت". بعد ذلك نزيد قيمة المتغير I بوحد لتصبح (٢) و بذلك نعرف أننا سنقوم بالطباعة للمرة الثانية. بعد ذلك نختبر قيمة I هل هي أكبر من ١٠ أي هل وصل عدد طباعة عبارة "أحسننت" عشر مرات أم لا. فإذا كانت الإجابة بلا فإن التفرع يعود ليكرر عملية الطباعة و زيادة قيمة العدد I بوحد. و هكذا تكرر عملية الطباعة طالما أن عدد التكرارات لم يصل إلى ١٠ مرات. فإذا تجاوزت قيمة المتغير I القيمة ١٠ فهذا يعني ان الشرط أصبحت قيمته بنعم و بذلك يتوقف البرنامج عن العمل.

ارسم مخططاً انسيابياً يقوم بقراءة مبلغ و يطبع قيمة زكاة هذا المبلغ. علماً بأن الزكاة بقسمة المبلغ على ٤٠.

الحل...

ارسم مخططاً انسيابياً يقوم بقراءة (١٥) مبلغ و يطبع قيمة زكاة هذا المبلغ. علماً بأن الزكاة بقسمة المبلغ على ٤٠.

الحل...

ارسم مخططاً يقوم بقراءة (١٥) درجة و يطبع متوسط الدرجات.

الحل...

ارسم مخططاً يقوم بقراءة (١٠) أعداد و يطبع مكعب كل عدد (مكعب العدد = العدد مضروب في نفسه ثلاث مرات).

الحل...

المتغيرات وأنواعها

المتغير هو رمز أو اسم نعرف الكمبيوتر به حتى يستخدمه الكمبيوتر لتخزين القيمة فيه و الفائدة من المتغيرات هي أن الكمبيوتر يخزن فيها القيم التي يقرأها من مستخدم البرنامج و التي يتم إدخالها من لوحة المفاتيح. الكمبيوتر يستطيع أن يخزن في هذه المتغيرات نوعين من القيم. و المتغيرات نوعان :

١- عديدي : هذا النوع من المتغيرات يخزن فيه الأرقام و الأعداد و هي إما أن تكون أعداد صحيحة أو حقيقية .

٢- حرفي : هذا النوع من المتغيرات يخزن فيه حرف من الحروف مثل "HJ ABC" . بما فيها الفراغ حيث انه يُعد نوع من أنواع الحروف و كذلك تدخل الحروف الخاصة مثل (& # %) المهم أن تكون بين علامتي تنصيص.

يمكن أن نعرف المتغيرات بأي اسم تريده ولكن يشترط مايلي:

١. أن لا يزيد اسم المتغير على ٤٠ حرفاً .

٢. يجب أن يكون أول حرف من الاسم حرفاً ألفاً بائياً أي الحروف من A إلى Z .

٣. يمكن استخدام الكلمات المحجوزة أو أحد أموامر برمجة لغة بيسك مثل (Print, Input, Read)

٤ يجب أن لا يحتوي اسم المتغير على فواصل أو مسافات أو رموز خاصة مثل : { ، @

إن الحاسب سوف يرى المتغيرات في البرنامج و لكنه كيف يستطيع أن يفرق بين انواع هذه المتغيرات. إنه لكي يستطيع الكمبيوتر من التفريق بين المتغيرات و ما يُخزن فيها هل هو رقم أو حرف أم هل هو رقم صحيح أو حقيقي فإننا نضع رمز معين بجانب اسم المتغير لكي نستطيع التفريق بينهم.

الرموز الخاصة التي تميز أنواع المتغيرات:

نوع المتغير	الرمز
عدد صحيح	%
عدد حقيقي	!
عدد حقيقي ذو دقة مضاعفة	#
متغير حرفي تسلسلي	\$

أمثلة على أنواع المتغيرات :

التعريف	نوع المتغير	السبب
X	متغير عددي	بدأ بحرف
YS	متغير حرفي	\$ بدأ بحرف وانتهى بعلامة
2X	ليس متغير	بدأ برقم
X2	متغير عددي	بدأ بحرف
NUMBER	متغير عددي	بدأ بحرف
THEN	ليس متغير	استخدامها كمتغير كلمة محجوزة لا يمكن
Z%	متغير عددي صحيح	\$ بدأ بحرف وانتهى بعلامة
"ALI"	ثابت حرفي	علامتي تنصيص ليس متغير لأنه موجود بين
NAMES	متغير حرفي	\$ بدأ بحرف وانتهى بعلامة

العمليات الحسابية

يمكن من خلال لغة بييسك برمجة العمليات الحسابية الأساسية و هي الضرب و القسمة و الجمع و الطرح. أما الأرقام ذات القوى العديدة مثل ٥ أس ٢ فإن لها طريقة كتابة خاصة و هي أن تكتب بهذه الطريقة ٢٨٥.

العملية	الرمز	مثال
الجمع	+	$Z = X + Y$
الطرح	-	$Z = X - Y$
الضرب	*	$Z = X * Y$

$Z = X / Y$	/	القسمة
$Z = X ^ 2$	^	الأس
$Z = P (A + B) / 2$	()	الأقواس

أمثلة على كيفية كتابة التعبيرات في بييسك

الآن سنتعرف على كيفية كتابة التعبيرات الحسابية بلغة البييسك

بييسك التعبير بلغة	الرياضي التعبير
$4 / (A + (C - B))$	$\frac{4}{A + (C - B)}$
$A ^ 3 + (C + 5)$	$A^3 + (C + 5)$

القواعد والأولويات التي يتبعها الحاسوب لتنفيذ أي تعبير حسابي :-

لغة البييسك لها طريقة في تنفيذ العمليات الحسابية و إيجاد النتيجة النهائية لها و ذلك حسب القواعد التالية:

أ . تنفذ العمليات داخل الأقواس بدأ بالأقواس الداخلية .

ب . تنفيذ عمليات الأس والتي يرمز بها بالرمز (^) .

ت . تنفيذ عمليات الضرب والقسمة

ث . تنفيذ عمليات الجمع والطرح

ج . عند وجود عمليات لها نفس مستوى الأولوية فإن العمليات تنفذ من اليسار إلى اليمين في البرامج المكتوبة باللغة الأجنبية.

مثال :-

ما هي قيمة المتغير X ؟

$$X = ((2 + 3 ^ 1 * 3 + 1) / 3 * 2 + 3) * (2 + 2 ^ 2 / 2)$$

$$X = ((2 + 3 * 3 + 1) / 3 * 2 + 3) * (2 + 4 / 2)$$

$$X = ((2 + 3 * 3 + 1) / 3 * 2 + 3) * (2 + 4 / 2)$$

$$X = ((12) / 3 * 2 + 3) * (2 + 4 / 2)$$

$$X = ((12) / 3 * 2 + 3) * (2 + 2)$$

$$X = (8 + 3) * (2 + 2)$$

$$X = 11 * 4$$

X= 44

التعابير المنطقية:

حتى يتضح القصد من التعابير المنطقية فإننا سنوضحها بهذا المثال. إذا كانت قيم المتغيرات $A=5$ و $B=6$ ثم قلت لك أن $A=B$ فهل منطقك أنت يقول خطأ. وهذا أيضاً يحصل مع الكمبيوتر حيث ينظر الكمبيوتر إلى أي تعبير منطقي و يعطيه قيمة صح أو خطأ. التعابير المنطقية لها عوامل سنناقشها في هذا الجدول:

العلاقة	مثال	العامل
A يساوي B	$A=B$	=
A لا يساوي B	$B \neq A$	$\neq$
A أصغر من B	$B > A$	<
A أكبر من B	$B < A$	>
A أصغر من أو يساوي B	$B \geq A$	$\leq$
A أكبر من أو يساوي B	$B \leq A$	$\geq$

و التعابير المنطقية التي سردنا أمثلة عليها في أعلاه هي تعابير منطقية بسيطة أي أنها تُقارن بين قيمتين، و لكن يوجد أيضاً تعابير منطقية مركبة و المقصود بالتعبير المنطقي المركب هو التعبير الذي يتكون من تعبيرين منطقيين بسيطين أو أكثر و بين كل تعبير بسيط و آخر رابط و هذه التعابير هي :

معنى الرابط	الرابط
هذا الرابط يصل بين تعبيرين منطقيين بسيطين ليكون تعبير منطقي مركب. هذا التعبير المنطقي المركب يكون صحيحاً إذا كان كلا التعبيرين المنطقيين البسيطين صحيحين.	AND
هذا الرابط يصل بين تعبيرين منطقيين بسيطين ليكون تعبير منطقي مركب. هذا التعبير المنطقي المركب يكون صحيحاً إذا كان أحد التعبيرين المنطقيين صحيحاً.	OR
هذا الرابط يصل بين تعبيرين منطقيين ليكون تعبير منطقي مركب. هذا التعبير المنطقي صحيحاً إذا كان أحد التعبيرين المنطقيين صحيحاً و ليس كلاهما صحيحين.	XOR
هذا العامل إذا سبق التعبير المنطقي سواء كان بسيطاً أو مركباً يقلب قيمته أو منطقته من صح إلى خطأ و من خطأ إلى صحيح.	NOT

أمثلة :

إذا كانت قيم المتغيرات كالتالي: $A=4, B=2, C=6, D=0$ فحدد هل التعابير المنطقية التالية صحيحة أم لا.

التعبير	نتيجة التعبير
$A=B$	خطأ
$(B) \text{ AND } (D=0 < A)$	صحيح
$< \text{OR } > (C > A)$	صحيح
$(A=4) \text{ XOR } (D=0)$	خطأ
$(A=4) \text{ XOR NOT } (D=0)$	صحيح

استناداً إلى قاعدة الأولويات في لغة بيسك ما هي نتيجة التعبير الحسابي التالي:

$$(5 + 3) / 2 + 2 ^ 2 \times 2$$

الحل خطوة بخطوة هو:

$$(8) / 2 + 4 \times 2$$

$$4 + 4 \times 2$$

$$4 + 8$$

$$12$$

ما هي قيمة المتغير Y إذا حسب التعبير الحسابي التالي:

$$Y = ((3 ^ 2 / 3 + 2 / 1) + 2 + 3) \times 2$$

الحل خطوة بخطوة هو:

$$Y = ((9 / 3 + 2 / 1) + (5) \times 2$$

$$Y = ((3 + 2) + (5) \times 2$$

$$Y = ((5) + (5) \times 2$$

$$Y = 10 \times 2$$

$$Y = 20$$

ما نتيجة التعبير الحسابي التالي:

$$X = ((2 + 3 ^ 1 \times 3 + 1) / 3 \times 2 + 3) \times (2 + 2 ^ 2 / 2)$$

الحل خطوة بخطوة هو:

$$X = ((2 + 3 \times 3 + 1) / 3 \times 2 + 3) \times (2 + 4 / 2)$$

$$X = ((12) / 3 \times 2 + 3) \times (2 + 2)$$

$$X = (8 + 3) \times (2 + 2)$$

$$X = 11 \times 4$$

$$X = 44$$

ما هي قيمة المتغير W إذا حسب التعبير الحسابي التالي:

$$w = (2 + 2 / 2) ^ 2 / (3 + 3) / 2$$

الحل خطوة بخطوة هو:

$$w = (2 + 1) ^ 2 (6) / 2$$

$$w = (3) ^ 2 / (6) / 2$$

$$w = 9 / 6 / 2$$

$$w = 0.75$$

حول المعادلة التالية إلى تعبير حسابي بلغة بييسك:

$$\frac{a+b}{3} - \frac{2^2+3}{7} + K$$

الحل:

$$(a+b)/3 - ((2^2=3)/7) + k$$

حول المعادلة التالية إلى تعبير حسابي بلغة بييسك ثم أوجد نتيجة المعادلة مستنداً على قاعدة الأولويات في لغة بييسك.

$$\frac{3+2}{5} + \frac{3^2 \times 2}{2+16} - 7$$

الحل:

$$M = ((3+2)/5) + ((3^2 \times 2)/(2+16)) - 7$$

$$= (5/5) + (9 \times 2)/(2+16) - 7$$

$$= (1) + (18)/(2+16) - 7$$

$$= 1 + 18/18 - 7$$

$$= 1 + 1 - 7$$

$$= 2 - 7$$

= - 5

إذا علمت أن قيمة $A=5$ و قيمة $B = 4$ فما هي نتيجة التعبير المنطقي التالي: $B \text{ and } B = A < A$

الحل: هذان تعبيرين منطقيين يربط بينهما الرابط and و حيث أن التعبير $B < A$ هو تعبير صحيح (T) و التعبير الثاني $B = A$ تعبير خاطئ (F) إذن فإن نتيجة التعبير كاملاً هو خطأ لأن $(F \text{ and } T)$.

ما هي قيمة التعبير المنطقي التالي هل هي صحيحة (T) أم خاطئة (F) حيث $A=7$ و $B=3$.

$(B > A \text{ AND } A > 5) \text{ XOR } (A = 10 \text{ OR } B = 3)$

الحل:

$(F \text{ AND } T) \text{ XOR } (F \text{ OR } T)$

$F \text{ XOR } T$

T

ما هي قيمة التعبير المنطقي التالي هل هي صحيحة (T) أم خاطئة (F) حيث $A=7$ و $B=0$ و $C = -3$ و $D = 3.5$.

$\text{NOT } (A > 3 \text{ AND } C > 0) \text{ XOR } (D > 3.90 \text{ OR } B > 0)$

الحل:

$\text{NOT } (F) \text{ XOR } (F)$

$T \text{ XOR } F$

F

ما هي قيمة التعبير المنطقي التالي هل هي صحيحة (T) أم خاطئة (F) حيث $A=3$ و $B=1.5$ و $C = 3.75$.

$(5 = 3) \text{ AND } (B = 3.6) \text{ XOR } (\text{NOT } (B < 2 \text{ AND } A = 3))$

الحل:

$(T \text{ AND } F) \text{ XOR } (\text{NOT } (T \text{ AND } T))$

$(F) \text{ XOR } (F)$

F

كيف تطبع المعلومات؟

إن الهدف من أي أغلب البرامج التي تُكتب هو الخروج بالنتائج أو المعلومات. فمثلاً إذا كان البرنامج مصمم ليجمع بين عددين فإن الهدف من هو طباعة حاصل جمع العددين. وفي لغة بيسك نستخدم أمر PRINT بغرض الطباعة و يقوم هذا الأمر بطباعة النصوص و نتائج العمليات الحسابية على الشاشة و كذلك طباعة محتويات المتغيرات أو الثوابت أو سطرًا خالياً على الشاشة. و التعابير التي يُمكن لأمر PRINT طباعتها هي:

نوع المطبوع	مثال
المتغير العددي	PRINT A
المتغير الحرفي	PRINT A\$
الثابت العددي	PRINT 6
الثابت الحرفي	PRINT "ALI"
حساب عملية حسابية ثم طباعة ناتجها (مزيج من القيم الثابتة)	PRINT 7 + 6
حساب عملية حسابية و التعويض عن قيم المتغيرات ثم طباعة ناتجها (مزيج من المتغيرات).	PRINT A + B
مزيج من المتغيرات و الثوابت يقوم بييسك بحساب هذا التعبير الراضي ثم يطبع الناتج فقط	PRINT (A + 5 * 2) / C
سطر خالي	PRINT

ملحوظات حول استخدام الأمر PRINT :

- يمكنك الاستعاضة عن الأمر PRINT بإشارة ؟
- يمكن طباعة أكثر من متغير أو تعبير باستخدام أمر PRINT واحد ولكن بشرط أن يفصل بينها بـ (؛ فاصلة منقوطة) مع عدم ترك مسافة بين التعبيرين أو المتغيرين. أو يفصل بينهم (، فاصلة) و ذلك بترك مسافة بين التعبيرين أو المتغيرين.

مثال (١) :

10 PRINT 5 ; 6 + 1

20 PRINT 5 , 6 + 1

النتائج :

7 5

7 5

مثال (٢) :

```
10 PRINT "WELCOME"
```

```
20 PRINT 6
```

```
30 PRINT 3 + 5
```

```
40 A = 9
```

```
50 T$= " NUMBER"
```

```
60 ABC = 82
```

```
70 PRINT ABC
```

```
80 PRINT "ABC"
```

```
90 PRINT A , 8
```

النتائج :

```
WELLCOME
```

```
6
```

```
8
```

```
NUMBER
```

```
82
```

```
ABC
```

```
9 8
```

ملاحظة:

يمكن الاستعاضة عن كناية أمر PRINT في البرنامج بوضع علامة استفهام ؟

مثال :

بدلا من كتابة

```
10 PRINT "WELCOME"
```

يمكن كتابة

```
10 ? "WELLCOME"
```

و سوف يكون نتيجة كلا الأمرين هو : WELCOME

كيف تقرأ البيانات؟

قبل التعرف على كيفية قراءة المتغيرات أو أن نشرح لماذا نحتاج إلى قراءة المتغيرات. لا يمكن لأي برنامج العمل مالم تمده بمتغيرات و حتى يستطيع تنفيذ بعض العمليات الحسابية على هذه المخلات و يخرج بالنتائج. و قد تكون هذه المتغيرات لها قيم مضمنة ضمن البرنامج و البعض الآخر من المتغيرات تحتاج إلى أن يقرأ من مستخدم البرنامج و بواسطة لوحة المفاتيح مثلاً. و لتنفيذ أمر القراءة نستخدم هذا الأمر لتمكين المستخدم من إدخال القيمة التي يريد و ذلك من خلال لوحة المفاتيح والتي مباشرة سوف يسند لها الكمبيوتر في المتغير سواء كان متغير عددي أو متغير حرفي.

وصيغته كالتالي :

المتغير INPUT

و يمكن قراءة أكثر من متغير باستخدام في سطر واحد

INPUT A,B,C

اتبع هذه القواعد عندما تستخدم أمر INPUT :

- يجب أن تكون عدد القيم المدخلة من مستخدم البرنامج كاستجابة للعبارة INPUT مساويا لعدد المتغيرات الموجودة في تلك العبارة .
- يجب أن يتوافق نوع كل قيمة مدخلة مع نوع المتغير الذي ستخزن فيه القيمة . فيجب إعطاء قيم حرفية للمتغير الحرفي وقيم عددية للمتغير العددي وهكذا .
- يجب أن تكون البيانات المدخلة من قبل المستخدم ثوابت عددية أو سلاسل حرفية ولا يجوز أن تكون على شكل عمليات أو معادلات .
- يجب أن تحاط القيم الحرفية بزوجين من علامات التنصيص في حالة وجود فراغات أو فواصل ضمن القيمة الحرفية حتى لا تعتبر أكثر من قيمة واحدة.
- يمكن إدخال أكثر من متغير بـ INPUT واحد وذلك بالفصل بينهم بفواصل.
- من الممكن استخدام رسائل توضيحية مع عبارة INPUT وذلك لنوضح ما نريد من المستخدم أن يدخله

مثال :

INPUT "ENTER YOUR NAME: ", NAMES

العبارة READ والعبارة DATA :

وهي شبيهة بجملة INPUT من حيث وظيفة قراءتها للبيانات حيث تقرأ المتغيرات من البرنامج إلا أن INPUT تقرأ البيانات المطلوبة من لوحة المفاتيح بواسطة المستخدم ولكن READ تقرأ البيانات من البرنامج نفسه والموجودة بعد DATA

ملحوظات حول استخدام READ و DATA :

- يمكن وضع عبارة DATA في أي مكان في البرنامج .

مثال :

READ A ,B ,C

Z = A + B + C

DATA 4 , 3 , 9

PRINT z

- يمكن قراءة أكثر من متغير بعبارة READ واحدة.

مثال :

READ A , B , K

- يمكن أن يكون في البرنامج أكثر من READ وأكثر من DATA ويمكن أيضا أن يكون في البرنامج READ واحدة وأكثر من DATA .
- يجب أن يكون عدد الثوابت في DATA يساوي عدد المتغيرات في READ أو أكثر

مثال :

READ A , B , C

DATA 4 , 6 , 7 , 3 , 20

- يجب توافق نوع المتغير في READ بالثابت في DATA

مثال :

READ A , B\$

DATA 4 , SAUD

- إذا كان الثابت الحرفي يحتوي على فاصلة أو مسافة فيجب أن يحاط بعلامتي تنصيص.

مثال :

READ A , B\$

DATA 6 , "A300,00"

ما الذي سيقوم به البرنامج التالي؟

READ A\$, B\$, C\$

DATA "Saudi", "Riyadh", "Rodah".

PRINT " ----- Information -----"

PRINT C\$,B\$, A\$

السطر الأول: سيقوم البرنامج في البداية بقراءة ثلاث قيم من داخل البرنامج و ليس من لوحة المفاتيح و لذلك استخدمنا أمر READ.

السطر الثاني: الأمر DATA سيقوم بتقديم القيم لأمر READ و هي ثلاثة قيم حرفية. حيث القيمة SAUDI سوف تُخزن في المتغير \$A و القيمة RIYADH ستخزن في المتغير \$B و القيمة RODAH ستخزن في المتغير \$C.

السطر الثالث: الأمر PRINT سيقوم بطباعة مابين علامتي التنصيص كما هي.

السطر الرابع: سيقوم بطباعة القيم المخزنة في المتغيرات على النحو التالي:

RODAH RIYADH SAUDI

أسئلة و تمارين

(١) اكتب برنامج يقوم بقراءة ثلاثة أعداد و يطبع هذه الأعداد بالعكس باستخدام أمر الإدخال INPUT.

الحل...

(١)

```
10 INPUT A, B, C
20 PRINT C, B, A
30 END
```

(٢) اكتب برنامج يقوم بقراءة ثلاثة أعداد و يطبع هذه الأعداد بالعكس باستخدام أمر الإدخال ...DATA READ.

الحل...

(

```
10 READ A, B, C
20 PRINT C, B, A
30 DATA 3, 75, 10
40 END
```

(٣) اكتب برنامج يقرأ درجات الثلاث الطلاب الأوائل في فصلك ثم يطبع درجة كل طالب في سطر و قبل كل اسم تكتب ترتيبه. و كمثال على المخرجات انظر للتالي:

الأول: ٩٩٩

الثاني: ٩٧٠

الثالث: ٩٦٥

الحل...

(٣)

```
10 INPUT A, B, C
20 PRINT "الأول:" ; A
30 PRINT "الثاني:" ; B
40 PRINT "الثالث:" ; C
50 END
```

(٤) اكتب برنامج يقوم بقراءة ثلاثة أعداد ثم يقوم بطباعة العدد الأول في العمود العاشر و العدد الثاني بعد العدد الأول بعشر خانات في نفس السطر، ثم يطبع العدد الثالث بعد سطرين.

الحل...

(٤)

```
10 INPUT A, B, C
20 PRINT TAB(10); A; SPC(10); B
30 PRINT
40 PRINT
50 PRINT C
```

التكرار

ماذا يعني التكرار في علم البرمجة، لن اشرح الكثير حتى أبين لك فائدة التكرار بل سأطلب منك كتابة برنامج و منه سوف أشرح معنى التكرار. اكتب برنامج يقوم بطباعة اسم علي خمس مرات. لا شك أنك ستقوم بكتابة البرنامج على النحو التالي:

```
10 Print "ALI"
20 Print "ALI"
30 Print "ALI"
40 Print "ALI"
50 Print "ALI"
```

لكن إن طلبت منك كتابة نفس البرنامج لطبع اسم "علي" مئة مرة فهل ستقوم بتكرار جملة الطباعة مئة مرة طبعاً هذا ليس بمنطق سليم لذلك وضع مفهوم التكرار في البرمجة. و سوف نناقش في دروس هذا الفصل كيفية استخدام أوامر التكرار.

حيث بواسطة أوام بسيطة تكرار أوامر معينة يستدعي الأمر منك تكرار تنفيذها عدداً معين من التكرارات أو تكرار أمر معين ضمن شرط معين. إن أوامر التكرار تفيدك في توفير جهد تكرار كتابة الأوامر.

جملة FOR...NEXT

تستخدم هذه الجملة لتكرار خطوة أو عدد من الخطوات في البرنامج بدلاً من أن تقوم بتكرار كتابتها و عليك أنت أن تحدد عدد مرات التكرار. الصيغة اللغوية لهذا التركيب هي :

FOR Counter = x TO n STEP m

-

تكرارها الجمل التي تريد

-

-

NEXT Counter

حيث أن :

القيمة الأولى للعداد	X
القيمة النهائية للعداد	n
تعبّر عن مقدار الزيادة التي ستزيدها في قيمة العداد.	STEP
قيمة الزيادة في كل دورة.	m
تعبّر عن نهاية الحلقة .	NEXT

مثال :

اكتب برنامج يدخل اسم ويطبعه ٧ مرات.

```
INPUT "ENTER YOUR NAME : ", N$
FOR S = 1 TO 7
PRINT N$
NEXT S
```

مثال :

اكتب برنامج يقوم بطباعة الأعداد من ١٥ إلى ١ .

```
FOR K= 15 TO 1 STEP -1
PRINT K
NEXT K
```

مثال :

اكتب برنامج يقوم بطباعة الأعداد الزوجية بين ٣٠ و ٩٠ .

```
FOR X = 30 TO 90 STEP 2
? X
NEXT X
```

مثال:

اكتب برنامجا يجمع الأعداد من ١ إلى ١٠ باستخدام الحلقة التكرارية باستخدام أمر FOR

```
10 S = 0
20 FOR I = 1 TO 10
30 S = S + I
40 NEXT I
50 PRINT "SUM OF THE NUMBERS ="; S
60 END
```

مثال:

اكتب برنامج يقوم بإدخال خمسة طلاب ودرجاتهم ومن ثم يطبع المتوسط الحسابي لمجموع الدرجات.

```
10 PRINT "ENTER ALL STUDENTS"
20 S = 0
30 FOR I = 1 TO 5
40 INPUT NAME$, MARK
50 S = S + MARK
60 NEXT I
70 AVG = S / I
80 PRINT AVG
```

الحلقات المتداخلة

يمكنك استخدام أمر FOR داخل البرنامج أكثر من مرة كل مادت الحاجة إلى ذلك، وكما رأينا في الأمثلة السابقة أن دالغ جملة FOR تضع أوامر قراءة وكتابة وعمليات إسناد (=) وعمليات حسابية. ولكن الذي سنتعلمه الآن أن تضع جملة تكرار FOR داخل جملة تكرار أخرى. وطبعاً استخدامك لهذه الطريقة يعتمد على حاجتك لها. و سنتضح فائدة هذه الطريقة من الأمثلة التي سنستعرضها.

أنظر للشكل العام لكتابة جملتي FOR متداخلتين.

```
FOR I = (القيمة الابتدائية) To (القيمة النهائية العظمى)
  FOR J = (الابتدائية القيمة) To (القيمة النهائية العظمى)
    -----
    -----
    ----- أواخر تنفيذها جملة التكرار الثانية الداخلية -----
    -----
  NEXT J
NEXT I
```

كما رأيت في الصورة العامة أن جملة التكرار الأولى التي تحتوي على العداد (I) تُكرر تنفيذ جملة التكرار الثانية. ففي الدورة الأولى يتم تنفيذ التكرار الثاني حتى ينتهي تماماً ثم يدور الدورة التالية من التكرار الأول. فمثلاً أنظر للمثال التالي:

```
FOR I = 1 To 5
  FOR J = 1 To 10
    PRINT "التكرار الثاني"
  NEXT J
```

NEXT I

التكرار الأول ينفذ التكرار الذي بداخله خمس مرات حيث أنه في الدورة الأولى أي عندما تكون $I=1$ يتم تنفيذ جملة التكرار الداخلية التي تقوم بالطباعة عشر مرات، عندما تنتهي يأتي وقت الدورة الثانية للتكرار الأول حيث تصبح قيمة $I=2$ حيث تبدأ جملة التكرار الداخلية الثانية بتكرار الطباعة عشر مرات ثم عندما تنتهي يأتي وقت الدورة الثالثة من التكرار الأول وهكذا. حتى في النهاية تتم طباعة جملة "التكرار الثاني" خمسون مرة.

انتبه

من الأخطاء الشائعة في استخدام جمل التكرار المتداخلة هو أن تكون على الصورة التالية:

FOR I = (القيمة الابتدائية القيمة) To (القيمة النهائية العظمى)
 FOR J = (القيمة الابتدائية) To (القيمة النهائية العظمى)

أوامر تنفذها جملة التكرار الثانية الداخلية

NEXT I
 NEXT J

لو تأملت الصورة السابقة لوجدت أن الحلقة الأولى تم إغلاقها قبل أن تنتهي الحلقة الثانية، حيث أنه لا يجوز أن تنتهي الحلقة الأولى إلا بعد أن تنتهي الحلقة الثانية الداخلية.

مثال:

أكتب برنامج يقوم بطباعة الأزواج المرتبة من واحد إلى ٣. و الأزواج المرتبة هي كما يلي:

1 1
 1 2
 1 3
 2 1
 2 2
 2 3
 3 1
 3 2

و هكذا حتى تنتهي الأزواج المرتبة.

الحل:

FOR I=1 TO 3
 FOR J = 1 TO 3

PRINT I ; J
 NEXT J
 NEXT I

جملة WHILE...WEND :

كما كلنا في مقدمة هذا الفصل أن التكرار قد يكون بعدد محدد أي معلوم مسبقاً مثل أن يقال كرر طباعة عبارة معينة خمسون مرة فعندها تستخدم جملة FOR . و أحياناً قد يكون التكرار غير معروف العدد بل يطلب منك تكرار تنفيذ أمر معين إذا كان شرط معين صحيح. مثال ذلك أن يُقال لك اكتب برنامج يقرأ مجموعة من الأعداد (غير معروف عددها) ثم يطبعها مضافاً إليها واحد، و يتوقف البرنامج عن قراءة الأعداد إذا كان العدد المقروء هو صفر. و لكتابة هذا البرنامج نستخدم أمر WHILE ...WEND الذي يقوم بتكرار أوامر معينة بعدد غير محدود طالما أن الشرط الذي يسمح باستمرار الدوران متحقق، كما ترى في حل المسألة السابقة.

```
10 INPUT A
20 WHILE A<>0
30 PRINT A+1
40 INPUT A
50 WEND
```

من المثال السابق تلاحظ أنه تم قراءة أول عدد من لوحة المفاتيح، ثم بواسطة جملة WEND يتم فحص القيمة المدخلة فإذا كانت القيمة تساوي صفر فإنه البرنامج لا يسمح للحاسب بالدخول في التكرار. أما إذا كانت القيمة لا تساوي صفرًا فإن البرنامج يسمح بدخول الحاسب للتكرار. حيث يضيف للقيمة الأولى واحد ثم يقوم بفحص القيمة هل تساوي صفر أم لا. فإذا كانت لا تساوي صفر فيسمح بإكمال التكرار و تنفيذ الأوامر التي بداخل جملة التكرار WHILE..WEND، أما إذا كانت القيمة المدخلة تساوي صفر فإن البرنامج سيتوقف عن عملية التكرار و يكمل ما بعد جملة التكرار من أوامر أي ما بعد كلمة WEND. و الصورة العامة لجملة WHILE...WEND هي كالتالي:

```
WHILE (تحققه حتى يتم التكرار الشرط الذي يجب)
-----
تكرارها الأوامر التي سوف يتم
-----
-----
WEND
```

انتبه

من المثال السابق تلاحظ أنه يوجد أمر قراءة واحدة INPUT قبل جملة WHILE..WEND و هذا ضروري حتى نوفر قيمة محددة ليتمكن أمر WHILE من اختبارها حتى يتقرر الدخول في التكرار من عدمه. فلو لم يكن هناك أمر قراءة قبل أمر WHILE فإن جملة WHILE لن تجد قيمة تقوم باختبارها.

أسئلة و تمارين

(١) اكتب برنامجاً يقوم بقراءة اسم واحد ثم يطبعه (٢٥) مرة.

الحل...

(١)

10 INPUT AS

```
20 FOR I=1 TO 25
30 PRINT A$
40 NEXT I
```

(٢) اكتب برنامجاً يقوم بطباعة الأعداد من ١٠ إلى واحد تنازلياً.

الحل...

(٢)

```
10 FOR I = 10 TO 1 STEP -1
20 PRINT I
30 NEXT I
```

(٣) اكتب برنامجاً يقوم بطباعة الأعداد الزوجية من واحد إلى عشرة.

الحل...

```
10 FOR I =2 TO 10 STEP 2
20 PRINT I
30 NEXT I
```

...

(٤) اكتب برنامجاً يقوم بطباعة الأعداد الفردية من واحد إلى عشرة.

الحل...

(٤)

```
10 FOR I =1 TO 10 STEP 2
20 PRINT I
30 NEXT I
```

(٥) اكتب برنامجاً يقوم بإيجاد مجموع الأعداد من واحد إلى عشرة.

الحل...

(٥)

```
10 SUM=0
20 FOR I=1 TO 10
30 SUM = SUM + I
40 NEXT I
50 PRINT SUM
```

(٦) اكتب برنامجاً يقوم بقراءة خمسة أعداد من لوحة المفاتيح ثم يوجد مجموعها و المتوسط الحسابي ثم يطبعهما. علماً بأن المتوسط الحسابي يُحسب عن طريق إيجاد مجموع الأعداد المدخلة ثم تقسم على عدد هذه الأعداد و هي الخمسة.

الحل...

(٦)

```
10 SUM =0
20 FOR I=1 TO 5
30 INPUT A
40 SUM= SUM + A
50 NEXT I
60 PRINT SUM/5
```

(٧) اكتب برنامج يقوم بقراءة عدد ثم يوجد مضروبه. مثلاً إذا كان العدد المُدخل هو ٤ فإن مضروبه هو :

4 x 3 x 2 x 1

الحل..

```
10 INPUT A
20 FACT =1
30 FOR I = 1 TO A
40 FACT = FACT * I
50 NEXT I
60 PRINT FACT
```

(٨) اكتب برنامج يقوم بطباعة الأعداد التالية حسب الشكل التالي:

1 2 3 4 5

2 3 4 5 6

3 4 5 6 7

الحل...

(

```
10 FOR I = 1 TO 3
20 FOR J = 1 TO I + 4
30 PRINT J;
40 NEXT J
50 PRINT
60 NEXT I
```

(٩) اكتب برنامجاً يقوم بقراءة مجموعة من الأعداد ثم يطبع مجموعها. و يتوقف قراءة الأعداد من لوحة المفاتيح عندما تكون القيمة المدخلة هي صفر.

الحل...

(٩

```
10 SUM = 0
20 INPUT A
30 WHILE A <> 0
40 SUM = SUM + A
50 INPUT A
60 WEND
70 PRINT SUM
```

(١٠) اكتب برنامجاً يقوم بقراءة أسماء طلاب و درجاته لأعمال السنة و درجة الاختبار الفصلي ثم يطبع مجموع درجاته. علماً بأن البرنامج لن يتوقف عن قراءة أسماء الطلاب إلا عندما تُدخل له كلمة "قف".

الحل...

(

```
10 INPUT N$, A, B
20 WHILE N$ <> "قف"
30 PRINT A + B
40 INPUT N$, A, B
50 WEND
```

التحكم في سير البرنامج

مفهوم التفرع

لقد علمنا من قواعد برنامج بيسك أن التنفيذ وفقا لأرقام السطور (أي بالتتالي) حسب ترتيب الأرقام تصاعديا. ولكن في بعض الأحيان يحتاج المبرمج إلى وسائل تمكنه من التحكم في خط سير البرنامج والتوجه لتنفيذ سطر محدد غير السطر المفترض تنفيذه حسب التسلسل. ويسمى ذلك بالتفرع حيث أن المبرمج يوجه الحاسب لكي يتفرع من التسلسل المتبع إلى سطر مختلف.

وهناك نوعان من التفرع :

١. تفرع غير مشروط.
٢. تفرع مشروط.

التفرع غير المشروط :-

وهو أن توجه الحاسب إلى أن ينتقل لسطر معين في البرنامج دون التقيد بشرط وذلك باستخدام أمر (GOTO) وتعني اذهب إلى. وصيغتها كالتالي :

رقم السطر المقصود GOTO

مثال (١) :

```
10 READ A , B
20 C = A + B
30 PRINT "A,B, هو : ",A," حاصل جمع"
40 GOTO 10
50 DATA 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8
60 END
```

يلاحظ في المثال السابق أن READ تبدأ في القراءة من أول قائمة البيانات في السطر المحتوي على البيانات DATA وبالترتيب و هو سطر رقم (٥٠). بمعنى أنه في أول مرة تتم فيها القراءة يجري إسناد القيم كالتالي A = 1 ، B = 2 ، ثم عندما تنفذ عبارة GOTO يذهب الحاسب إلى السطر (١٠) ليقوم بالقراءة ويتم الإسناد مرة أخرى كالتالي : A = 3 ، B = 4 و هكذا حتى تنتهي البيانات ، عندما تنتهي الأرقام المقروءة في أمر Data في السطر (٥٠) سوف تحصل على الرسالة Data Out Of أي البيانات غير كافية ويتوقف تنفيذ البرنامج.

مثال(٢):

```
10 INPUT "أدخل رقماً صحيحاً", X
20 PRINT X , X^2
30 GOTO 10
40 END
```

ملحوظة : لا ينصح باستخدام الأمر GOTO في كتابة البرامج ، لأنه يؤدي إلى بناء برامج غير هيكليّة من الصعب تتبع العمليات فيها ، وبالتالي يصعب تصحيحها واكتشاف الأخطاء بها وصيانتها بسبب فقدان هيكليتها.

التفرع المشروط باستخدام IF .. THEN

وهو أن توجه الحاسب إلى أن ينتقل من سطر إلى سطر آخر في البرنامج ولكن عندما يتحقق شرط معين . الشكل العام لعبارة [IF .. THEN] هو :

(تنفيذ أمر) THEN (الشرط) IF)

معنى ذلك إذا صح الشرط فسوف تنفذ العملية المطلوبة، كمثال على ذلك إذا كان عمرك أكبر من ١٨ سنة فسوف تعطى رخصة قيادة . فنلاحظ أنه يتم أولاً التأكد من صحة الشرط . فإذا كانت نتيجة الشرط صحيحة فإن البرنامج ينفذ العملية الموجودة بعد كلمة THEN أما إذا كان الشرط ليس صحيحاً فإنه لا يتم تنفيذ العملية التي تلي THEN . ويمكن بعد THEN كتابة رقم السطر المراد التوجه لتنفيذه.

مثال (١):

10 INPUT A	10 INPUT A
20 IF A > 10 Then 30	20 IF A > 10 THEN PRINT "OK"
30 Print "OK"	30 END
40 END	
هنا لم نضع عملية تنفذ مباشرة بعد THEN وإنما تم وضع رقم السطر المراد التوجه لتنفيذه وكلا الحالتين متكافئتين	إذا كانت قيمة A أكبر من ١٠ فإنه سوف ينفذ العملية الموجودة بعد THEN وهي طباعة كلمة OK

والآن سوف نعيد كتابة المثال (١) الموجود في التفرع الغير المشروط الذي ذكرناه سابقاً ولكن سوف نتفادى ظهور رسالة الخطأ Out Of Data ونجعله ينتهي بصورة طبيعية باستخدام التفرع المشروط .

```
10 READ A , B
20 IF A = 0 THEN 70
30 C = A + B
40 PRINT " هو ", B, " و ", A, " حاصل مجموع العدد"
50 GOTO 10
60 DATA 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 0 , 0
70 END
```

البرنامج سوف يعمل حسب فكرة البرنامج السابق و لكننا قيمتين هما صفر و صفر في نهاية الأرقام الموجودة ضمن أمر DATA في السطر (٦٠) و هي تدل على أن القيم المراد حسابها قد انتهت و لمعرفة ما إذا وصلنا لنهاية الأرقام أضفنا شرطاً واحداً يقوم بالتأكد من أن القيمة المقروءة تساوي صفراً فإذا كانت $A = 0$ فهذا يعني أن أخذت جميع القيم التي تخصها و وصلت إلى القيمة صفر أي أن الأرقام الموجودة مع الأمر DATA قاربت على الانتهاء و حتى لا نحصل على

رسالة الخطأ Out Of Data نقوم بنقل سير البرنامج إلى السطر (٧٠) و هو أمر END ليتوقف عمل البرنامج.

THEN . . IF لصورة أخرى

حيث نقوم الحاسب بتنفيذ عبارة من عبارات بييسك بعد THEN مباشرة.

مثال (١):

```
10 INPUT A , B
20 IF A > B THEN PRINT " A > B "; A
30 IF B > A THEN PRINT "B > A "; B
40 IF A = B THEN PRINT " A = B "
50 END
```

هذا البرنامج يستخرج القيمة الكبرى من قيمتين يتم قرأتهم من لوحة المفاتيح وفي حالة تساويهما يطبع عبارة تُفيد أن القيمتين متساويتين، و نفس الطيقة إذا كانت قيمة أكبر من الأخرى.. فلو قمنا مثلاً بتنفيذ هذا البرنامج و أدخلنا في الحالة الأولى قيمتين هما ٥ و ٧

```
? 5 , 7
B > A 7
```

فلو أعدنا تنفيذ البرنامج بمدخلات جديد هما ٦ و ٩ :

```
? 6 , 9
A > B
```

مثال (٢):

هذا المثال يستخرج العدد الأكبر من ضمن عشرة أعداد يتم إدخالها للبرنامج :

```
10 INPUT A
20 MAX = A
30 FOR I = 1 TO 9
40 INPUT A
50 IF A > MAX THEN MAX = A
60 NEXT I
70 PRINT "القيمة العظمى"; MAX
80 END
```

صورة متقدمة للتفرع المشروع

ELSE . . IF . . THEN لصورة عبارة

IF [الشرط] THEN [جواب الشرط ١]

ELSE [جواب الشرط ٢]

و هذه صورة متقدمة لأمر IF . ويزيد في هذه العبارة عن سابقتها الجزء ELSE و للتوضيح نقول إن عبارة IF تقوم باختبار شرط تحدده أنت حسب حاجة البرنامج فإذا كان الشرط صحيحاً فإن الحاسب ينفذ جواب الشرط الأول الذي بعد عبارة THEN و قد يكون جواب الشرط عملية حسابية أو أمر GOTO ، أما إذا كان الشرط غير صحيح فإن الحاسب ينفذ جواب الشرط الثاني الذي بعد عبارة ELSE و قد يكون جواب الشرط عملية حسابية أو أمر GOTO .

لو أردنا ترجمة عبارة الشرط هذه فسوف نقول:

إذا الشرط صحيح إذن نفذ الجواب الأول

و إلا نفذ الجواب الثاني

مثال (١)

10 INPUT A

20 IF A > 10 THEN PRINT "A > 10" ELSE PRINT "A < 10"

30 END

بالنظر للمثال السابق تجد أن البرنامج يقوم بقراءة عدد من لوحة المفاتيح و يضعه في المتغير A ثم يقوم باختبار شرط معين و هو التأكد من أن القيمة المخزنة في المتغير A أكبر من عشرة وذلك بالمقارنة بعلامة أكبر من (>). إذا كانت القيمة المخزنة في A أكبر من (١٠) فسوف ينفذ الحاسب الشرط الأول الذي بعد كلمة THEN و هو طباعة عبارة "A < ١٠". أما إذا كانت نتيجة الشرط خاطئة فسوف يقوم البرنامج بتنفيذ الأمر الذي بعد كلمة ELSE و هي طباعة عبارة "A > ١٠".

التفرع المشروط باستخدام GOTO .. ON :

هناك أسلوب آخر للتفرع الغير مشروط و هو GOTO .. ON وذلك في حالة وجود أكثر من اختيارين تريد تنفيذهم . وصيغتها كالتالي :

رقم سطر آخر ، ... ، رقم سطر ، رقم سطر ON X GOTO

حيث تقوم هذه العبارة بالنظر إلى قيمة المتغير X فإذا كانت قيمته لرقم السطر الأول فيتم الانتقال إلى هذا السطر أما إذا كانت قيمة المتغير X تساوي رقم السطر التالي فيتم الانتقال إليه.

مثال (٢):

10 READ X

20 ON X GOTO 30 , 50 , 70

30 PRINT "السطر رقم 30"

40 GOTO 10

50 PRINT "السطر رقم 50"
60 GOTO 10
70 END
80 DATA 1, 2, 3

في السطر (١٠) يقوم البرنامج بقراءة قيمة و يضعها في المتغير X ثم في السطر (٢٠) يقوم البرنامج بفحص قيمة X بواسطة عبارة ON X GOTO ... فإذا كانت قيمة X تساوي (٣٠) فإن البرنامج سيذهب البرنامج إلى لينفذ الأمر الموجود في السطر (٣٠). أما إذا كانت قيمة X تساوي (٥٠) فإن البرنامج سيذهب لينفذ ما هو موجود في السطر (٥٠) وهكذا. لو ترجمنا معنا عبارة ON X GOTO إلى:

إذا قيمة X تساوي ٣٠ إذهب إلى السطر ٣٠ و نفذ مافية.

إذا قيمة X تساوي ٥٠ إذهب إلى السطر ٥٠ و نفذ مافية.

إذا قيمة X تساوي ٧٠ إذهب إلى السطر ٧٠ و نفذ مافية.

لاحظ أن البرنامج في السطر (٤٠) و السطر (٦٠) يأمر الحاسب بالذهاب إلى السطر (١٠) وذلك لقراءة قيمة جديدة من القيم الموجودة في عبارة DATA .

حل أسئلة و تمارين

(١) بين الخطأ في كل من الأسطر التالية من بييسك :

IF A THEN 60

صيغة عبارة IF هي أن يكون بعدها شرط معين إذا تحقق يتم تنفيذ العملية التي بعد THEN و هذا المثال لا يوجد به شرط (مقارنة) بعد IF بل متغير عددي.

IF A = B THEN B * B - 4 * A * C

قلنا بعد THEN يتم تنفيذ عملية أو توجيه الحاسب إلى سطر معين وهنا لم تكتمل العملية بعد THEN لعدم وجود متغير يتم إسناد القيمة الناتجة من العملية إليه، أي أنه لابد أن تكون العبارة بعد على الشكل التالي:

IF A = B THEN Y = B * B - 4 * A * C

IF A = G\$ THEN 60

هنا يوجد خطأ وهو محاولة مقارنة متغير من نوع عددي A مع متغير من نوع حرفي وهذا لا يصح.

B THEN 60 >< IF A

هنا يوجد خطأ في طريقة كتابة الشرط أو المقارنة و هو علامة العلاقة $B >< A$ والصحيح أن يكون $B <> A$

(٢) اكتب برنامج يحسب قيمة Y في المعادلة التالية :

Y=

على البرنامج أن يطلب من المستخدم إدخال قيمة X ، ثم يتأكد من القيمة المدخلة التي يجب أن تقع ضمن الحدود المسموح بها وهي ($X > 10$ الحل..)

ج - ٢

```
10 INPUT X
20 IF X > -10 AND X < 10 THEN 30 ELSE END
30 Y = SQR (100 - ( X ^ 2 ))
40 PRINT Y
50 END
```

(٣) اشرح عمل البرنامج التالي ، ثم أعد كتابته مع محاولة التخلص من أمر GOTO واستبدالها بأوامر أخرى. الحل ..

```
10 INPUT X
20 IF X < 3 THEN 40
30 GOTO 60
40 PRINT "TYPE 1"
50 GOTO 70
60 PRINT "TYPE 2"
70 END
```

ج - ٣

هذا البرنامج يطلب من المستخدم إدخال قيمة X ومن ثم يقارن هذه القيمة إذا كانت اصغر من 3 يقوم بطباعة بتوجيه الحاسب إلى السطر (40) ليقوم بطباعة عبارة TYPE1 إما إذا كانت قيمة X اكبر من 3 فيقوم بطباعة عبارة TYPE2، وبذلك يصبح البرنامج بعد التخلّص من أمر GOTO و استخدام عبارة IF كالتالي:

```
10 INPUT X
20 IF X < 3 THEN PRINT "TYPE 1" ELSE PRINT "TYPE 2"
30 END
```

(4) أعد كتابة البرنامج التالي باستخدام ON .. GOTO

```
10 IF A = 1 THEN 500
20 IF A = 2 THEN 600
30 IF A = 3 THEN 700
40 PRINT " ERROR"
```

الحل..

ج - 4

لا بد من فهم معنى البرنامج حيث انه إذا كانت قيمة A=1 فسوف يذهب البرنامج إلى السطر 500 وإذا كانت A=2 فيذهب إلى السطر 600 أما إذا كانت قيمة A=3 فيذهب إلى السطر 700

الحل :

```
10 ON A GOTO 500 , 600 , 700
20 PRINT "ERROR"
```

(5) اكتب برنامج يطلب من المستخدم إدخال عدد صحيح ، ثم يقوم بتوجيه البرنامج إلى الأسطر 100، 300، 400، 500 إذا كان العدد المدخل يساوي 11، 12، 13، 14 على الترتيب . الحل..

ج - 5

```
10 INPUT X
20 IF X = 11 THEN 100
30 IF X = 12 THEN 300
40 IF X = 13 THEN 400
50 IF X = 14 THEN 500
```

(٦) اكتب برنامجا يطلب من المستخدم إدخال أي من عددين بين ١ إلى ٥ . ثم يقوم البرنامج بكتابه العدد بالأحرف ، أي يكتب "واحد" إذا كان الرقم المدخل = ١ ، وهكذا.

الحل..

ج -٦

```
10 INPUT "ENTER NUMBER BETWEEN 1 TO 5 : ", A
20 IF A >= 1 AND A <= 5 THEN 30 ELSE END
30 IF A = 1 THEN PRINT "واحد"
40 IF A = 2 THEN PRINT "اثنان"
50 IF A = 3 THEN PRINT "ثلاثة"
60 IF A = 4 THEN PRINT "اربعة"
70 IF A = 5 THEN PRINT "خمسة"

80 END
```

(٧) اكتب برنامجا يقوم باستخراج القيمة الكبرى (العظمى) لخمس أعداد مقروءة مستخدما جملة READ-DATA ومفترضاً أن هذه الأعداد هي ٦ ، ١٥ ، -١٠ ، صفر ، ٨

الحل..

ج -٧

```
10 READ X
20 MAX = X
25 FOR I = 1 TO 5
30 READ X
40 IF X > MAX THEN MAX = X
50 NEXT I
60 DATA 6 , 15 , -10 , 0 , 8
70 PRINT "The maximun value : "; MAX
80 END
```

(٨) اكتب برنامج يقوم باستخراج القيمة الصغرى لعشرة أعداد مدخلة .

الحل..

ج -٨

```
10 INPUT A
15 MIN = A
20 FOR I = 1 TO 10
30 INPUT A
40 IF A < MIN THEN MIN = A
50 NEXT I
60 PRINT A
70 END
```

(٩) اكتب برنامج يقوم بإدخال قيمة ثم يحدد نوعها "القيمة موجبة" أم "القيمة سالبة" أم "تساوي صفر".

الحل..

ج - ٩

```
10 FOR I = 1 TO 10
20 INPUT X
30 IF X > 0 THEN PRINT "القيمة موجبة"
40 IF X < 0 THEN PRINT "القيمة سالبة"
50 IF X = 0 THEN PRINT "تساوي صفر"
60 NEXT I
70 END
```

(١٠) أعد حل السؤال السابق مفترضاً أن القيمة عندما تساوي الصفر فإنها تكون موجبة واستخدم في برنامجك عبارة
ELSE ... IF ... THEN

الحل..

ج - ١٠

```
10 FOR I = 1 TO 10
20 INPUT X
30 IF X >= 0 THEN PRINT "القيمة موجبة" ELSE PRINT "القيمة سالبة"
40 NEXT I
70 END
```

الدوال

والدالة هي عبارة عن برنامج صغير مكتوب مسبقاً و محفوظ داخل لغة بيسك يمكن استدعائه من خلال برنامجك والاستفادة منه ، إضافة إلى ذلك يمكنك أنت أن تكتب دالة وتستدعيها أكثر من مرة داخل برنامجك . والدوال في بيسك تنقسم إلى:

١. دوال رياضية (عددية)
٢. دوال حرفية
٣. دوال تحويل
٤. دوال معرفة من قبل المبرمج.

ونلاحظ أن الثلاثة أنواع الأولى أنها دوال داخلية ضمن اللغة، أي تأتي معرفة في لغة بيسك و مبرمجة مسبقاً لكي تستخدمها مباشرة. أما النوع الأخير فيتم تعريفه بواسطة المبرمج، أي بواسطتك أنت.

الدوال الرياضية

الجدول التالي يبين الدوال الرياضية في بيسك .

شرح الدالة	الدالة في بيسك	الدالة الرياضية
حساب الجذر التربيعي للعدد X	SQR (X)	
القيمة المطلقة للعدد X	ABS (X)	
اللوغاريتم الطبيعي للعدد X	LOG (X)	Log X
الدالة الأسية للعدد X	EXP (X)	
دالة الجيب للزاوية X	SIN (X)	Sin x
دالة جيب التمام للزاوية X	COS (X)	Cos x
دالة الظل للزاوية X	TAN (X)	Tan x
دالة عكس الظل للزاوية X	ATAN (X)	X
دالة توليد رقم عشوائي للعدد X	RAN (X)	

أمثلة على استخدام الدوال الرياضية :

مثال (١): اكتب برنامجاً يطلب من المستخدم إدخال عشرة أعداد ومن ثم يطبع الجذر التربيعي لها .

10 FOR I=1 TO 10

20 INPUT X

30 PRINT SQR (X)

40 NEXT I

50 END

مثال(٢): اكتب برنامجا يولد عشرون عددا عشوائيا حيث يكون مجال الأعداد من الصفر إلى العشرة.

10 RANDOMIZE

20 FOR K = 1 TO 20

30 X = RAN * 10

40 PRINT X

50 NEXT K

60 END

مثال(٣): اكتب برنامجا يدخل زاوية بالتقدير الستيني ويحسب جيب الزاوية وجيب تمام الزاوية.

10 INPUT A

20 X = A * 3.14 /180

30 ? SIN (X)

40 ? COS (X)

50 END

من البرنامج تلاحظ أنه يجب تحويل الزاوية إلى التقدير الدائري أولا قبل التعامل معها في البرنامج.

الدوال الحرفية :

طريقة استخدامها نفس طريقة استخدام الدوال الرياضية إلا أنها تتعامل مع البيانات الحرفية .

وظيفة	الدالة الحرفية
إيجاد الحرف المناظر للرقم في جدول الترميز أسكي	CHR\$ (m)
إيجاد ترميز أسكي لأول حرف في سلسلة الحروف x\$	ASC(x\$)
لاستخلاص عدد m من الحروف من الجملة x\$ بدءا من اليسار	LEFT\$ (x\$, m)
لاستخلاص عدد m من الحروف من الجملة x\$ بدءا من اليمين	RIGHT\$ (x\$, m)
لاستخلاص جزء طوله n بدءا من الحرف رقم m من الجملة x\$	MID\$(x\$, m , n)

LEN(X\$)	لحساب طول الجملة الحرفية
----------	--------------------------

أمثلة على استخدام الدوال الحرفية :

مثال (١)

10 INPUT A	10 INPUT AS
20 X\$=CHR\$(A)	20 X = ASC(AS)
30 PRINT X\$	30 PRINT X

في الجدول السابق استخدمنا الدالتين هما دالة ASC التي تقوم بتحويل الحرف إلى قيمة عددية تدل على رقمه في جدول الحروف. و الدالة CHAR\$ التي تعطيها رقماً من جدول الحروف ثم تخرج لك الحرف المناظر لهذا الرقم من الجدول.

في المثال الأول الأيسر قرأ البرنامج رقماً و وضعه في المتغير A ثم في السطر الثاني وضع المتغير A وسط الدالة CHAR\$ التي ستخرج قيمة الحرف المناظر لهذا الرقم من جدول الحروف ثم وضع الحرف المخرج في المتغير الحرفي X\$. و في السطر الثالث طبع الحرف المُخزن في المتغير X\$.

في المثال الثاني الأيمن ابتداء البرنامج بقراءة حرف وضعه في المتغير AS ثم في السطر الثاني استخدم الدالة ASC() التي تقوم بحساب الرقم المناظر للحرف المُخزن في المتغير AS و تضع الرقم في المتغير X و في السطر الأخير نطبع قيمة X و هو الرقم المناظر للحرف المدخل من بداية البرنامج.

مثال (٢):

```
10 INPUT X$
20 PRINT LEFT$(X$,4)
30 PRINT RIGHT$(X$,3)
40 PRINT MID$(X$,2,3)
50 PRINT LEN$(X$)
```

بالنظر للبرنامج السابق نجد أننا:

- في البداية قرأنا قيمة حرفية و وضعناها في متغير حرفي هو X ثم استخدمنا الدالة Left\$ التي تساعدنا في أخذ أول أربعة حروف ابتداء من الجهة اليسرى للقيمة الحرفية المخزنة في المتغير X و طبعنا قيمة الدالة Left\$.
- بعد ذلك استخدمنا الدالة Right\$ التي تساعدنا في أخذ أول ثلاثة حروف ابتداء من الجهة اليمنى للقيمة الحرفية المخزنة في المتغير X و طبعنا قيمة الدالة Right\$.
- بعد ذلك استخدمنا الدالة MID\$ التي تساعدنا في أخذ جزء من القيمة الحرفية المخزنة في المتغير X ابتداء من الحرف الثاني الموجود في الجهة اليسرى و بطول ثلاثة حروف. و طبعنا قيمة الدالة MID\$.
- بعد ذلك طبعنا قيمة الدالة LEN\$(X) و التي تقوم بحساب عدد الحروف للقيمة الحرفية المخزنة في المتغير X\$.
- لاحظ أنه قبل كل دالة حرفية البرنامج وضعنا أمر الطباعة Print حيث يقوم الحاسب بحساب قيمة الدالة الحرفية ثم يقوم بطباعة النتيجة التي قامت الدالة الحرفية بإخراجها.

المخرجات هي كالتالي بفرض أن القيمة المدخلة هي QBASIC

? QBASIC

QBAS

SIC

BAS

6

الدوال المعرفة بواسطة المستخدم :

وهي الدوال الغير معرفة داخليا في لغة بيسك . بل تقوم أنت بتعريفها داخل البرنامج ومن ثم استدعائها في أي مكان في البرنامج وفي بعض الأحيان أكثر من مرة . فبدلاً من كتابة مجموعة من الأوامر المحددة بلغة بيسك لإنتاج قيمة معينة تقوم بكتابة هذه الأوامر مرة واحدة في الدالة التي عرفتتها فقط بدلاً من كتابة الأوامر في كل مرة تحتاج إلى تنفيذها . وطريقة تعريف الدالة المبرمجة كما يلي :

تعريف = (معاملات) اسم الدالة FN DEF

أسئلة وتمارين

(١) باستخدام دوال بيسك احسب جيب زاوية قدرها ٦٠ درجة. الحل..

10 F = 60 * 3.14 / 180

20 PRINT SIN (F)

لاحظ أنه قبل التعامل مع الزوايا باستخدام دوال بيسك الحسابية لابد من تحويل الزاوية من التقدير الدائري (٦٠ ، ٩٠ ، ١٢٠ ..) إلى التقدير الستيني وذلك بضرب الزاوية في ٣,١٤ ثم نقسم الناتج على ١٨٠ . ثم يُمكن تطبيق دوال بيسك الحسابية على الزاوية كما في الحل أعلاه.

(٢) اكتب برنامجاً في بيسك يولد عشرين عدداً عشوائياً تقع قيمها فيما بين ١ إلى ١٠٠ ثم يقوم البرنامج بتحديد أكبر عدد فردي من بين هذه الأعداد العشوائية وطباعة العدد . الحل..

10 RANDOMIZE

20 Max=0

30 FOR K = 1 TO 20

40 X = RAN * 100

50 IF (X / 2) = INT (X / 2) THEN 70 ELSE 60

60 IF X > MAX THEN MAX = X

70 NEXT K

80 PRINT MAX

90 END

و حيث أننا سوف نستخدم دالة RAN التي تقوم بتوليد ارقام عشوائية فإنه لابد من إعلام الحاسب بذلك و ذلك ما فعلناه في السطر الأول حيث قمنا بكتابة أمر RANDMOIZE و الذي يُخبر الحاسب بأننا سنستخدم دالة RAN. في السطر الثاني فرضنا أن المتغير MAX يساوي صفر و هذا المتغير نقوم بتخزين أعلى قيمة فيه و لذلك فرضنا أن أعلى قيمة هي الصفر حتى الآن و لو بدى لنا فيما بعد أن هناك قيمة أعلى من الصفر فسوف نُخزنها في المتغير MAX.

في السطر الثالث استخدمنا أمر التكرار FOR الذي سيساعد البرنامج على تكرار قراءة عشرين رقم و المقارنة فيما بي هذه الأرقام.

في اسطر الرابع بعد أمر FOR استخدمنا دالة RAN لتخرج لنا رقماً يقع بين ١ و ١٠٠ حيث كتبنا $RAN * 100$ ثم نُسند النتيجة التي سوف تُخرجها الدالة RAN في المتغير X

في السطر (٥٠) نقوم باختبار العدد المخزن في X و الذي جاءنا من الدالة RAN لمعرفة ماذا كان عدداً فردياً أم لا.

و لفهم طريقة المقارنة في السطر خمسين نفرض أن الرقم الذي حصلنا عليه من الدالة RAN هو (٧) فلو قسمنا ٧ على العدد ٢ فإن الناتج سيكون ٣,٥ ثم لو قسمنا العدد (٧) على (٢) ثم أخذنا القيمة الصحيحة و أهملنا القيمة العشرية و يساعدنا في ذلك الدالة INT و قارنا بين القيمتين نجد أنهما غير متساويتين مما يدل على أن هذا العدد فردي لأن أي عدد فردي نقسمه على (٢) يكون ناتج قسمته عدد عشرياً (بفاصلة) أما العدد الزوجي فعندما نقسمه على (٢) يكون الناتج بدون ليس عدداً عشرياً بل عدداً صحيحاً. لاحظ كيف نكتب الشرط في السطر (٥٠) بعد ما نعوض عنه بالأرقام:

٦٠ IF (7/2) = INT(7/2) Then 70 Else

IF 3.5 = 3 Then 70 Else 60

لكن لو كان العدد الذي جاءنا من الدالة RAN هو عدداً زوجياً مثل (٦) فإنه السطر (٥٠) يصبح:

٦٠ IF (6/2) = INT(6/2) Then 70 Else

IF 3 = 3 Then 70 Else 60

إذا كان ناتج المقارنة في السطر (٥٠) صحيحاً أي عدد زوجياً فإن هذا الرقم لا يعيننا و يجب القفز لقراءة الرقم الذي يليه و لذلك وجهنا البرنامج إلى السطر (٧٠) أما إذا كان الشرط خاطئاً فننتقل للسطر (٦٠).

في السطر (٦٠) نقوم بمقارنة العدد الفردي الذي حصلنا عليه من RAN و المخزن في المتغير X لنعرف هل هو أكبر من القيمة المخزنة في MAX لأننا دائماً نفترض أن المتغير MAX يحتوي على أكبر قيمة. فإذا كان العدد المخزن في X أكبر من العدد المخزن في MAX فيجب تخزين هذا العدد في MAX لأن المتغير MAX يجب أن يحتفظ بالقيمة الفردية الأكبر طوال فترة تنفيذ البرنامج.

بعد تنفيذ التكرار (٢٠) مرة الذي يحتوي على السطر ٤٠ و ٥٠ و ٦٠ و هي الأسطر المسنولة عن عملية القراءة من الدالة RAN و استخراج العدد الفردي الأكبر نخرج من التكرار و قد خزننا أكبر عدد فردي في المتغير MAX ثم نقوم بطباعته في السطر ٨٠.

(٣) اكتب برنامجاً يطلب من المستخدم أن يدخل عدداً موجباً أو سالباً ، ثم يحسب الجذر التربيعي للعدد ويطبع النتيجة على الشاشة . فإذا كان العدد المدخل سالباً فإنه يطبع الجذر التربيعي متبوعاً بالحرف (خ) للدلالة على أن القيمة الناتجة قيمة خيالية .

الحل..

```
10 INPUT A
20 IF A >= 0 THEN PRINT SQR(A)
30 IF A < 0 THEN PRINT "خ"
40 END
```

(٤) ما هي الدوال المعرفة من قبل المستخدم ؟ وما هي فائدتها؟ الحل..

هي الدوال غير المعرفة داخليا في لغة بيسك بل يقوم المبرمج بتعريفها وكتابتها داخل برنامجها.

وفائدتها أنها يمكن استخدامها في البرنامج أكثر من مرة ويقوم المبرمج بتعريفها داخل برنامجها لتسهيل الاستخدام والتعامل معها مثل أي دالة من دوال بيسك .

(٥) عرف دالة اسمها مساحة الدائرة وتسوي (ط X نق ٢) ، حيث ط = النسبة التقريبية (٣,١٤) ، و (نق) هي نصف قطر الدائرة . ثم اكتب برنامجا يطلب من المستخدم إدخال نصف القطر ، ويقوم بحساب مساحة الدائرة باستخدام الدالة المعرفة ، ويطبع النتيجة على الشاشة . الحل..

```
10 DEF FNCRCL (X) = 3.14 * X ^ 2
20 INPUT A
30 C = FNCRCL (A)
40 PRINT C
```

لا يوجد في لغة بيسك دالة تقوم بحساب مساحة الدائرة لذلك لابد أن تقوم أنت بتعريف دالة من عندك تقوم بحساب مساحة الدائرة، وهذا ما قمنا به في السطر (١٠) وذلك باستخدام أمر DEF ومعناه "عرف". بعد ذلك نقوم بقراءة عدد في السطر (٢٠) ثم نحسب قيمة مساحة العدد المُخزن في A باستخدام الدالة FNCRCL (A) التي ستأخذ القيمة المخزنة في A ثم تُعوض عنها مكان X في جسم الدالة CRCL.

المصفوفات (المنظومات)

كنا نقول أن المتغير يشير إلى قيمة واحدة، فمثلاً عندما نقول المتغير X فهذا يعني أنه يمكن أن نخزن فيه قيمة عددية صحيحة واحدة فقط في نفس الوقت بينما المصفوفة عبارة عن مجموعة من القيم التي يشار إليها بنفس المتغير ، فيمكن أن ننشئ مصفوفة اسمها X مكونة من ١٠ عناصر (خانات) تقبل أن نخزن فيها عشرة أرقام صحيحة و لتوضيح طريقة عمل المصفوفات أنظر للرسم أدناه مثلاً:

١٠	٩	٨	٧	٦	٥	٤	٣	٢	١
١٧	٤	١٠	٥	٥٥	٦	٢٣	٩	٥	١١

المصفوفة X

في الرسم أعلاه رسم توضيحي لمصفوفة اسمها X تتكون من عشرة خانات كل خانة مخزن بها رقم صحيح. من اليمين يوجد العنصر الأول ثم العنصر الثاني وهكذا. لكن كيف يمكن التعامل مع عنصر معين في المصفوفة. سيتضح هذا من الأمثلة التالية:

- عندما نقول (X(1 فإننا نعني العنصر الأول من المصفوفة X و هذا العنصر مخزن فيه عدد هو ١١.
- عندما نقول (X(4 PRINT فإن الحاسب سوف يقوم بطباعة القيمة المخزنة في العنصر رقم ٤ و هي ٢٣.

و بالطبع يمكن أن نخزن في المصفوفة قيم عددية أو قيم حرفية حسب التعريف الذي حددته للمصفوفة. و لكن قيم المصفوفة تكون من نوع واحد فقط أي لا بد أن تكون جميعها عددية أو جميعها حرفية في نفس الوقت.

متى نستخدم المصفوفة في البرنامج؟

إن اختيار استخدام المصفوفة في البرنامج يعتمد على المطلوب منك في البرنامج فإذا طلب منك مثلاً قراءة درجة ثلاثين طالب فإنك كما اعتدت في السابق ستقوم بكتابة البرنامج التالي:

```
10 For I = 1 To 30
20 READ X
30 NEXT I
```

لاحظ أنك كلما قراءت درجة طالب و خزنتها في المتغير X ثم قرأت درجة طالب آخر فإنك ستفقد الدرجة التي قبلها. و لكي تحتفظ بدرجات الثلاثين طالباً في نفس الوقت عندها يجب أن تستخدم المصفوفات التي توفر لك متغيراً من نوع مصفوفة رقمية به (٣٠) خانة.

تعريف المصفوفة :

قبل استخدام المصفوفة في البرنامج يجب أن تعرف متغير من نوع مصفوفة و تحدد نوع البيانات التي تخزنها في هذه المصفوفة الجديدة هل هي رقمية أم حرفية. يمكن أن نعرف المصفوفة كمصفوفة عددية أو مصفوفة حرفية والفرق بينهما أن المصفوفة العددية يمكن أن نخزن فيها أعداداً فقط ونتعامل مع محتواها كأعداد مثال :

$$X(2) = 3 + 1$$

أما إذا كانت مصفوفة حرفية فإنه يتم تخزين متغيرات حرفية مثال :

AS(3) = "ALI"

حسناً...لو أردنا تعريف مصفوفة أسمها X مكونة من ١٠٠ عنصر يتم فيها تخزين قيم عددية و مصفوفة حرفية يتم فيها تخزين أسماء ١٠٠ طالب فإننا نقوم بالتالي في بداية البرنامج:

DIM X(100)

DIM AS(100)

إذن أمر DIM يساعدك في تعريف المصفوفات. و تذكر أنه إذا كانت عدد عناصر المصفوفة التي ستقوم بتعريفها (١٠) عناصر أو أقل فإنه لك مطلق الحرية أن تعرف المصفوفة في بداية البرنامج أو لا.

مثال(١)

اكتب برنامجا يطلب من المستخدم إدخال أسماء ودرجات خمسة طلاب ويقوم بتخزين الأسماء في مصفوفة حرفية و الدرجات في مصفوفة عددية أخرى.

```
10 FOR I= 1 TO 5
20 INPUT NAMES ( I ) , G ( I )
30 NEXT I
```

لا حظ أننا عندما نقوم بتعبئة المصفوفة بالبيانات لا بد من استخدام أمر التكرار FOR الذي سيقوم بتغيير قيمة مؤشر العناصر (هو بين القوسين التي بعد اسم المصفوفة). فبدلاً من أن نكتب البرنامج بهذه الطريقة البدائية التالية:

```
10 INPUT NAMES(1), G(1)
20 INPUT NAMES(2), G(2)
30 INPUT NAMES(3), G(3)
40 INPUT NAMES(4), G(4)
50 INPUT NAMES(5), G(5)
60 END
```

فإن استخدام التكرار سوف يوفر عليك في عدد الأسطر التي تكتبها حيث أن التكرار سيغير قيمة المؤشر في كل دورة يدورها.

مثال (٢):

أكمل البرنامج السابق لكي يقوم باستخراج أعلى درجة من درجات الطلبة الخمسة ثم اطبع اسم الطالب.

```
40 MAX=0
50 FOR I= 1 TO 5
60 IF G(I) > MAX THEN 70 ELSE 90
70 MAX = G(I)
80 MNAME$ = NAMES (I)
90 NEXT I
100 PRINT NAME$, MAX
110 END
```

في البداية فرضنا أن أكبر قيمة حتى الآن هي صفر و أنها مُخزنة في المتغير MAX الذي لا نخزن فيه سوى أكبر درجة

فقط. بعد ذلك استخدمنا التكرار للبحث في مصفوفة الدرجات عن درجة طالب تكون أكبر من القيمة المخزنة في المتغير MAX (الذي نفترض أن أكبر قيمة مخزنة فيه). وذلك بالمقارنة الموجودة في السطر (٦٠) فلو اكتشفنا أن هناك عنصراً في مصفوفة الدرجات أكبر من القيمة المخزنة في MAX التي اعتبرها أكبر قيمة فيجب عمل الآتي:

١. في السطر (٧٠) نقوم بتخزين باستبدال الرقم الموجود في المتغير MAX بالقيمة التي اكتشفنا أنها أكبر.
٢. في السطر (٨٠) نقوم بالاحتفاظ بنسخة من اسم الطالب صاحب الدرجة الأعلى في متغير حرفي هو NAMES. تذكر أن الطالب صاحب الدرجة المخزنة في مصفوفة الدرجات ورقم خاتمه هي (٣) مثلاً فإن اسمه ستجده في الخانة رقم (٣) أيضاً من مصفوفة الأسماء.

أسئلة و تمارين

- (١) اكتب برنامجاً يطلب من المستخدم إدخال عشرين عدداً ثم يقوم بحفظ هذه الأعداد في المصفوفة W .

الحل..

- (٢) اكتب برنامجاً يقرأ ١١ مجموعة من البيانات . وكل مجموعة تتكون من اسم الشخص متبوعة بعمرة . ثم يقوم البرنامج بحفظ هذه البيانات في مصفوفتين . إحداهما للأسماء والأخرى للأعمار . استخدم العبارات READ و DATA في هذا البرنامج.

الحل..

- (٣) اشرح العملية التي يقوم بها البرنامج التالي :

```
10 DIM X( 50 )
20 FOR I=1 TO 50
30 X ( 0 ) = 0
40 NEXT I
50 END
```

الحل..

- (٤) اكتب برنامجاً يقوم بإدخال أسماء عشرة طلاب في مصفوفة ودرجاتهم لإحدى المواد في مصفوفة أخرى . ثم في مصفوفة ثالثة اجعل البرنامج يخزن تقديرات هؤلاء الطلاب بناء على هذه الشروط :

إذا كانت الدرجة أكبر أو تساوي ٩٠ يكون التقدير A .

إذا كانت الدرجة أكبر من أو تساوي ٨٠ و أصغر من ٩٠ فإن التقدير B .

إذا كانت الدرجة أكبر من أو تساوي ٧٠ و أصغر من ٨٠ فإن التقدير C .

إذا كانت الدرجة أكبر من أو تساوي ٦٠ و أصغر من ٧٠ فإن التقدير D .

إذا كانت الدرجة أصغر من ٦٠ فإن التقدير F .

الحل..

(١)

```
10 DIM X(20)
20 FOR I= TO 20
30 INPUT X(I)
40 NEXT I
50 END
```

(٢)

```
10 DIM NAMES$( 11 )
20 DIM AGE( 11 )
30 FOR J = 1 TO 11
40 READ NAMES$ ( J ) , AGE( J )
50 NEXT J
60 DATA "FAHD", 11 , "ANAS" , 13 , ..... , "SAAD", 17
70 END
```

لاحظ أنه في السطر (٦٠) قمنا بوضع بجانب الأمر DATA اسم الطالب الأول و درجته ثم اسم الطالب الثاني و درجته و هكذا حتى اكملنا أنهينا احدى عشر اسماً.

(٣) في هذا البرنامج عرفنا مصفوفة عددية مكونة من خمسين عنصراً اسمها X . قمنا باستخدام أمر التكرار FOR لتعبئة كل عنصر من عناصر المصفوفة بقيمة صفر.

(٤)

```
10 DIM NAMES$ ( 10 )
20 DIM COURSE (10)
30 DIM G$ ( 10 )
40 FOR I= I TO 10
```

```

50 INPUT NAMES$ ( I ) , COURSE ( I )
60 NEXT I

70 FOR P = 1 TO 10
80 IF COURSE ( P ) >= 90 THEN G$ ( P ) = "A"
90 IF COURSE ( P ) >= 80 AND COURSE ( P ) < 90 THEN G$ ( P ) = "B"
91 IF COURSE ( P ) >= 70 AND COURSE ( P ) < 80 THEN G$ ( P ) = "C"
92 IF COURSE ( P ) >= 60 AND COURSE ( P ) < 70 THEN G$ ( P ) = "D"
93 IF COURSE ( P ) < 60 THEN G$ ( P ) = "F"
95 NEXT P
96 PRINT "الدرجة" , "التقدير" , "الاسم"
97 PRINT "-----"

98 FOR K = 1 TO 10
99 PRINT K , NAMES$ ( K ) , COURSE ( K ) , G$ ( K )
100 NEXT K
110 END
    
```

شرح البرنامج:

- في السطر ١٠ و ٢٠ و ٣٠ قمنا بتعريف ثلاثة مصفوفات الأولى خاصة بأسماء الطلاب و الثانية خاصة ب درجات الطلاب و الثالثة خاصة بتقديرات الطلاب.
- من السطر ٤٠ حتى ٦٠ قمنا بتعبئة مصفوفة الأسماء و الدرجات بالبيانات التي نقرأها من لوحة المفاتيح.
- من السطر ٧٠ حتى ٩٥ استخدمنا أمر التكرار FOR لنقوم بالمرور على كل درجة طالب في مصفوفة الدرجات COURSE لنحدد تقديره حسب درجته و ذلك بمقارنة الدرجة مع كل شرط من الشروط الموجودة لدينا و حسب الدرجة نقوم بتخزين التقدير في المصفوفة G.
- و حيث أننا سنطبع تقديرات الطلاب من الأفضل طباعة ترويسة في رأس التقرير و هذ ما قمنا به في السطر ٩٦ و ٩٧ .
- من السطر ٩٨ حتى ١٠٠ نقوم باستخدام أمر التكرار لطباعة محتويات المصفوفات.

البرامج الفرعية

البرنامج الفرعي هو مجموعة من أوامر بييسك التي تعلمتها سابقاً لتكون برنامج صغيراً يؤدي وظيفة محددة. تكتب مرة واحدة ويمكن استخدامها أكثر من مرة داخل البرنامج بدل من تكرار كتابته.

أهمية البرامج الفرعية:

تحتوي معظم البرامج الكبيرة على عمليات متكررة في أجزاء مختلفة من البرنامج . فمثلاً قد تحتاج إلى تقوم بتعبئة بيانات في مصفوفة أو طباعة بيانات موجودة في مصفوفة أكثر من مرة فبدلاً من كتابة الأسطر التي تقوم بهذا الشئ أكثر من مرة في البرنامج فيكفي أن تكتبها مرة واحدة في برنامج فرعي ثم تنادي هذا البرنامج متى ما أردت لتوفر على نفسك الوقت و الجهد.

فوائد البرامج الفرعية

- توفر الوقت و الجهد.
- نقلل أسطر البرنامج.
- تجعل درجة تعقيد البرنامج أقل

- تساعد في تعديل أخطاء البرنامج بسهولة.

التعامل مع البرامج الفرعية:

لكي نتعرف على طريقة عمل البرامج الفرعية راقب هذا البرنامج و حاول فهم طريقة عمله.

```
10 FOR I=0 TO 8
20 READ A(I)
30 NEXT I
40 GOSUB 100
50 PRINT " المتوسط الحسابي " ;AVG
60 DATA 10 , 15 , 20 , 25 , 30 , 10 , 40 , 30 , 5
70 END
```

```
100 S = 0
110 FOR I = 0 TO 8
120 S = S + A ( I )
130 NEXT I
140 AVG = S / 9
150 RETURN
```

هذا البرنامج يقوم بقراءة درجات (٨) طلاب و يضعها في مصفوفة ثم يوجد المتوسط الحسابي لدرجات الطلاب حيث يوجد مجموع درجات الطلاب و يقسمه على عدد الطلاب. لقد بدأ البرنامج الرئيسي من السطر ١٠ و حتى السطر ٧٠ حيث إنتهى بعبارة END . و بدأ البرنامج الفرعي من السطر (١٠٠) و حتى السطر (١٥٠). لاحظ أن البرنامج في البداية قام بتعبئة مصفوفة (A) بدرجات الطلاب ثم في السطر (٤٠) قام بتنفيذ البرنامج الفرعي حيث استخدم أمر GOSUB 100 و معناها أذهب إلى البرنامج الفرعي الذي يبدأ من السطر (١٠٠).

في البرنامج الفرعي عرفنا متغير اسمه (S) سوف نضع فيه المجموع، و لكي نوجد المجموع استخدمنا أمر التكرار في السطر (١١٠) لنجمع كل درجة موجودة في خانات المصفوفة على المتغير (S) و بشكل تراكمي. و عندما ينتهي التكرار سوف نخرج بمجموع درجات الطلاب مُخزنة في المتغير S.

في السطر (١٤٠) حسبنا المتوسط الحسابي و نضعه في المتغير AVG . ثم بعد ذلك لا بد من العودة إلى البرنامج الرئيسي الذي تفرعنا منه بناءً على أمر GOSUB السابق و للعودة استخدمنا أمر RETUTRN و معناها (إرجع) . بعد تنفيذ هذا الأمر سوف يعود البرنامج من حيث أتى ليكمل بقية البرنامج الرئيسي.

لتنفيذ البرنامج الفرعي من داخل البرنامج الرئيسي نستخدم GOSUB

أسئلة وتمارين

(١) بين وجه الخطأ في البرنامج التالي :

```
10 REM ERROR
20 Y = 2
30 GOSUB 100
```

```
40 PRINT Y
100 Y = 45
110 END
```

الحل..

(١) الخطأ أننا استخدمنا في هذا البرنامج GOSUB للتوجه إلى البرنامج الفرعي ولم نعد للبرنامج الرئيسي لأننا لم نستخدم RETURN .

(٢) أكتب برنامجاً مكوناً من برنامج رئيس وثلاثة برامج فرعية لتنفيذ مايلي :

- يبدأ البرنامج بتقبل ثلاثة أعداد من المستخدم .
- يستخرج البرنامج أصغر عدد من بين الأعداد المدخلة .
- يطبع قيمة أصغر عدد ، ومربع ذلك العدد ، ومكعب العدد .

الحل..

(٢)

```
10 PRINT "Enter the numbers : "
20 GOSUB 100
30 GOSUB 150
40 GOSUB 200
50 END

100 FOR I = 1 TO 3
110 INPUT X(I)
120 NEXT I
130 RETURN

150 MIN = X(1)
160 FOR J = 1 TO 3
170 IF X(J) < MIN THEN MIN = X(J)
180 NEXT J
190 RETURN

200 PRINT MIN
210 PRINT MIN ^ 2
220 PRINT MIN ^ 3
250 RETURN
```

شرح البرنامج:

البرنامج الرئيسي يبدأ من السطر (١٠) و حتى السطر (٥٠) يتم فيه نداء البرامج الفرعية بواسطة أمر GOSUB .

ينتقل التنفيذ من البرنامج الرئيسي إلى البرنامج الفرعي الأول الذي يبدأ من السطر (١٠٠) و حتى السطر (١٣٠) و فيه نقوم بتعبئة المصفوفة X بثلاثة أرقام بعد قراءتها من لوحة المفاتيح رقماً رقماً، بعد ذلك يعود التنفيذ للبرنامج الرئيسي بواسطة أمر RETUTRN .

ينتقل التنفيذ من البرنامج الرئيسي إلى البرنامج الفرعي الثاني الذي يبدأ من السطر (١٥٠) و حتى السطر (١٩٠). حيث يقوم أولاً بافتراض أن أصغر رقم في المصفوفة هو الأول حيث يُخزنه في المتغير MIN و الذي يعني الأصغر. ثم يستخدم أمر التكرار لينتقل بين عناصر المصفوفة باحثاً عن رقم يكون أصغر من الرقم المُخزن في MIN . و عملية اكتشاف الرقم الأصغر تتم باستخدام المقارنة الموجودة في السطر (١٧٠). بعد ذلك يعود التنفيذ للبرنامج الرئيسي بواسطة أمر RETUTRN .

ينتقل التنفيذ من البرنامج الرئيسي إلى البرنامج الفرعي الثالث الذي يبدأ من السطر (٢٠٠) و حتى السطر (٢٥٠). حيث يقوم بطباعة قيمة المتغير MIN و الذ تم فيه تخزين أصغر قيمة تم اكتشافها. ثم بعد ذلك يطبع أصغر قيمة بعد رفعها للقوة (٢) أي تربيع أصغر قيمة. ثم بعد ذلك يطبع أصغر قيمة بعد رفعها للقوة (٣) أي تكعيب أصغر قيمة.

(٣) باستخدام البرامج الفرعية اكتب برنامجاً يدخل أسماء عدد من الطلبة ودرجات المواد الدراسية (افترض أربع مواد) والمطلوب هو حساب المتوسط الحسابي للمواد التي درسها كل طالب ، وكذلك معرفة الطالب المتفوق في كل مادة أي الحاصل على أعلى درجة ، وأيضا الطالب الذي حصل على أعلى متوسط حسابي .

الحل..

(٣)

```
10 INPUT "HOW MANY STUDENTS", Z
11 GOSUB 90
20 FOR I = 1 TO Z
30 INPUT STUD$(I)
40 INPUT S1, S2, S3, S4
50 GOSUB 100
55 GOSUB 200
60 NEXT I
65 MAXAVG = 0
70 GOSUB 300
80 GOSUB 400
85 END

90 MAX1 = 0
92 MAX3 = 0
93 MAX3 = 0
94 MAX4 = 0
```

95 SUM = 0

96 RETURN

100 SUM(I) = S1 + S2 + S3 + S4

120 AVG(I) = SUM(I) / 4

150 RETURN

200 IF S1 > MAX1 THEN 201 ELSE 210

201 MAX1 = S1

202 MAXNAME1\$ = STUD\$(I)

210 IF S2 > MAX2 THEN 211 ELSE 220

211 MAX2 = S2

212 MAXNAME2\$ = STUD\$(I)

220 IF S3 > MAX3 THEN 221 ELSE 230

221 MAX3 = S3

222 MAXNAME3\$ = STUD\$(I)

230 IF S4 > MAX4 THEN 231 ELSE 250

231 MAX4 = S4

232 MAXNAME4\$ = STUD\$(I)

250 RETURN

300 FOR K = 1 TO Z

310 IF AVG(K) > MAXAVG THEN 320 ELSE 340

320 MAXAVG = AVG(K)

330 MAXNM\$ = STUD\$(K)

340 NEXT K

350 RETURN

400 PRINT "THE BEST STUDENT IN COURSE 1 IS : "; MAXNAME1\$

410 PRINT "THE BEST STUDENT IN COURSE 2 IS : "; MAXNAME2\$

420 PRINT "THE BEST STUDENT IN COURSE 3 IS : "; MAXNAME3\$

430 PRINT "THE BEST STUDENT IN COURSE 4 IS : "; MAXNAME4\$

440 PRINT

450 FOR P = 1 TO Z

460 PRINT "STUDUDENT NAME : "; STUD\$(P); " COURSES AVG = "; AVG(P)

470 NEXT P

480 PRINT

450 PRINT "THE Best student gets MAX AVG is :"; MAXNM\$

460 RETURN

(٤) اكتب برنامجا يقوم بإدخال أسماء عدد من الطلبة في مصفوفة ودرجاتهم في مصفوفة أخرى ويطبع اسم الطالب الحاصل على أكبر درجة . ثم يطبع أسماء ودرجات الطلبة الذين أسماؤهم تبدأ بالأحرف ABDUL .

الحل..

(٤)

```
10 INPUT Z
20 FOR I=1 TO Z
30 INPUT STUD$(I), M(I)
40 NEXT I
50 MAX = 0
60 GOSUB 100
70 PRINT MAXNAMES
80 GOSUB 200
90 END

100 FOR J = 1 TO Z
110 IF M(J) > MAX THEN 120 ELSE 140
120 MAX = M(J)
130 MAXNAMES$ = STUD$(J)
140 NEXT J
150 RETURN

200 FOR K = 1 TO Z
210 IF LEFT$(STUD$(K), 5) = "ABDUL" THEN 220 ELSE 250
220 PRINT STUD$(K), M(K)
250 NEXT K
260 RETURN
```

