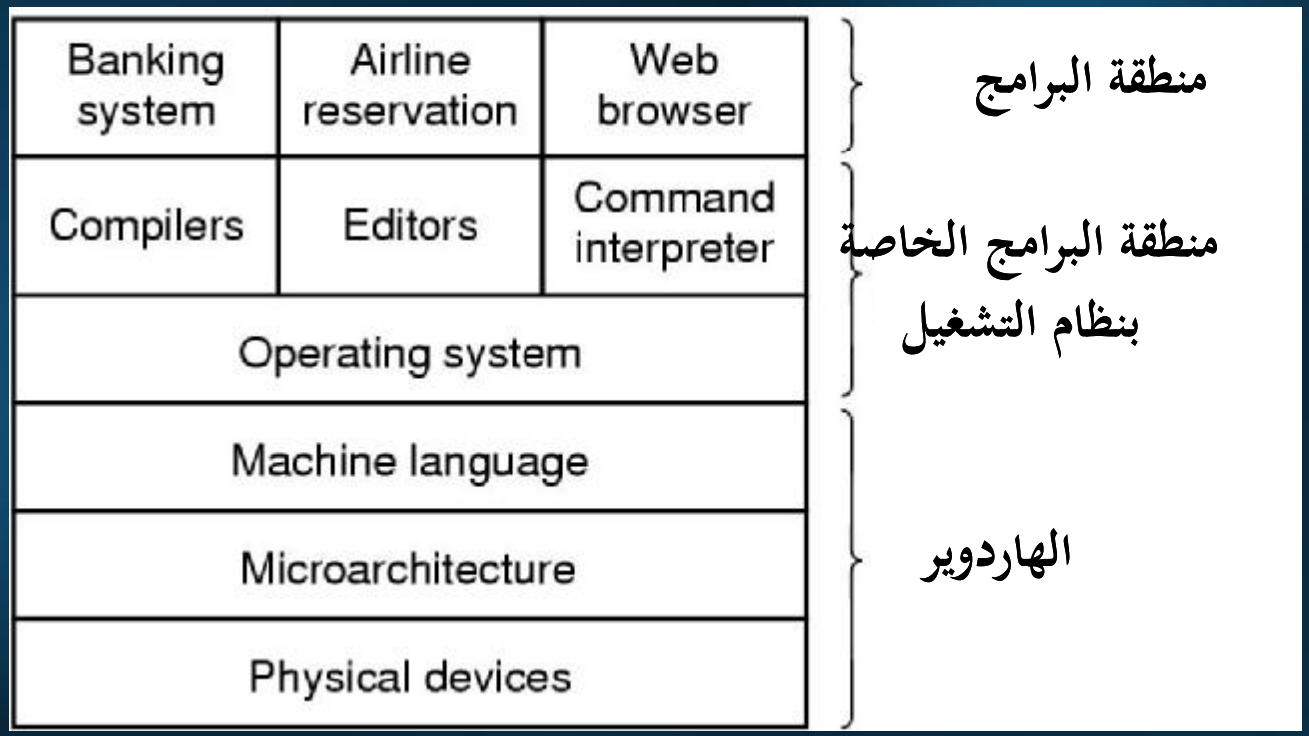
ما المقصود بالبرمجة ؟

- برمجة الحاسوب : هى عملية تزويد الكمبيوتر بالأوامر اللازمة لتنفيذ مهمة معينة بطريقة معينة ...
- البرمجة: هي عملية يتم من خلالها تحديد كيفية التعامل مع البيانات المدخلة الى الكمبيوتر للحصول على النتائج المرجوة.

أهمية البرمجة

- تكمن أهمية البرمجة في عالم الحاسوب إلى أنها تسهل التعامل مع ال Hardwareمن خلال قيامها بدور الوسيط بين المستخدم و المكونات المادية.



كيف يعمل الكمبيوتر؟

الكمبيوتر عبارة عن مجموعة مترابطة من الدوائر الكهربائية. تخيّلوا معي زر كهربائي يشغّل عدداً من المصابيح التي إمّا تكون مطفأة أو مشتعلة. لذا، لإ إن الكمبيوتر يفهم رقمين، هما (صفر) و (واحد). و يقوم الكمبيوتر بتحويل كل شيء إلى ما يقابله حسب نظام العد الثنائي (يعني ١ و ٠) و تسمّى هذه اللغة (التي تتكون من هذين الرقمين فقط) بلغة الآلة (Machine Code).

لغات البرمجه

- لغات عالية المستوى High –Level
- لغات منخفضة المستوى Low-Level
  ١- Machine Code
  ٢- Assembly

مثال(جمع عديين)

١- Machine Code

١- كتابة العد الاول ٤٢

٢-كتابة العدد الثانى ٢٤

٣-جمع العد الاولو العدد الثانى

النتيجه:

- ١٠١٠١١١٠١٠٠٠١٠١٠٠١٠١٠٠١٠١٠٠١٠١٠٠١٠١٠٠٠
  ١٠٠١٠١١٠١٠٠١٠٠١١١١٠٠١١١٠٠٠٠١٠١٠١٠١
  ٠١٠١٠١٠٠٠١٠٠١٠٠٠٠٠٠٠١٠٠١٠١١١١٠١١١٠١٠٠١
  ٠١٠١٠١٠١

لغات الجيل الاول

- و كما ترون فإنه من الصعب جدّاً فهم شيئاً كالذي ترونه بعقلنا البشري. لذا فقد كان من الصعب أيضاً ايجاد الأخطاء و تصحيحها. ولكن تلك التي تعمل، فأنها تعمل بسرعة شديدة جدّاً نظراً لأنها مكتوبة بلغة تفهمها الآلة بصورة مباشرة. و تسمّى البرامج المكتوبة بلغة الآلة ببرامج الجيل الأول.
- لم يستطع أجدادنا المبرمجون الاستمرار على هذا الحال، فقد كان فعلاً صعباً، لذت قاموا باختراع لغة التجميع (Assembly) و هي لغة أبسط من لغة الآلة نظراً لاحتوائها على أوامر مثل (ADD و MOV)، و هكذا كما ترون فقد ابتعدنا قليلاً عن لغة الآلة التي سبق و ركّزت على أنها لا تفهم سوىالصفر والواحد. و هنا ظهرت الحاجة الماسّة إلى المفسّر (Interpreter) و هو برنامج يقوم بتحويل الأوامر بالتتابع من لغة ال Assembly إلى لغة الآلة، و يوقف البرنامج فوراً في حالة مواجهة خطأ في البرنامج و لا يقوم يالنظر (حتى لجبر الخاطر) إلى بقية البرنامج :-)
- نلاحظ هنا أننا حصلنا على برنامج تسهل كتابته نسبياً و يمكن بسهولة ايجاد الأخطاء فيه و تصحيحها

مثال(جمع عددين)

Assembly٢ٍ

```
MOV AX,42
MOV BX,24
ADD CX,AX
ADD CX, BX
```

- لكن.. التحويل من لغة Assembly إلى لغة الآلة، يأخذ جزء من وقت البرنامج، و هكذا أصبح البرنامج أبطأ قليلاً، و حين أقول قليلاً، فأنا أعني جزء من آلاف الأجزاء من الثانية الواحدة. و تعتبر لغة التجميع من الجيل الثاني (الذي يحتوي على هذه اللغة فقط)
- المبرمجون لم يعجبهم الحال لذا أرادوا لغات تكون قريبة من لغة الإنسان، يفهمها بسهولة و ليست معقّدة مثل لغة التجميع. لذا قاموا بوضع العديد من لغات البرمجة التي تقوم على أوامر مباشرة بلغتنا البشرية التي نفهمها بسهولة، ففي لغة الباسكال مثلاً إذا أردنا طباعة كلمة (mama) على الشاشة نكتب الأمر التالي ضمن البرنامج:
- writeln("mama")

لغات عالية المستوى High –Level

- سميت بهذا الاسم لأنه اصبح بامكان المبرمج كتابة البرامج دون معرقة تفاصيل كيفية قيام الحاسب بهذه العمليات ، كمواقع التخزين و تفاصيل الجهاز الدقيقه.
- تعبيرات اللغات ذات المستوى العالي شبيهه الى درجة كبيرة باللغة التي يستخدمها الانسان في التخاطب و التواصل مع الاخرين.
- تتميز بسهولة اكتشاف الاخطاء و تصحيحها و يمكن تشغيلها على اكثر من جهاز. كما يمكن استخدام أكثر من لغة برمجه على جهاز واحد.

- يجدر بالذكر أن جميع لغات السابقة إذا تمّت ترجمتها على نظام تشغيل، لا تعمل إلا على هذا النظام.
- مثلا إذا قمنا بكتابة برنامج بلغة السي و قمنا بعمل ترجمة له (Compilation) على نظام التشغيل ويندوز، فان هذا البرنامج من المستحيل بأي حال من الأحوال أن يعمل على نظام الماكنتوش. ما نفعله في هذا الحالة أننا نأخذ البرنامج و نضعه على جهاز ماكنتوش و نقوم بترجمة مرة أخرى.
- إذاً أي برنامج يتصرّف هكذا:
- البرنامج --------> المفسّر/المترجم --------> لغة الآلة
- Compiler /interpreter
- ولأن المترجم متصل مباشرة كما ترون بلغة الآلة فإنه يعتمد عليها. و يختلف من نظام تشغيل إلى آخر.

بعض أنواع لغات البرمجة الشائعة الاستخدام

- ١. لغة البيسك BASIC Language ولغة فيجوال بيسك Visual Basic
- وهي لغات بسيطة عامة الأغراض وسهلة التعلم ويستخدمها المبتدئون في جميع الأعمال، وخاصة في التطبيقات العلمية، وهي اختصار للمعنى Beginners All-Purpose Symbolic Instruction Code ولبساطة هذه اللغة واستخدامها في التعليم ظهرت لها عدة إصدارات منها BASICA, GWBASIC, TURBO BASIC, QUICK BASIC.
- كما ظهرت أيضاً لغة فيجوال بيسك (البيسك المرئي) Visual Basic وهي لغة برمجة مرئية وتعتبر لغة مطورة من لغة البيسك وهي خاصة لإنتاج برمجيات ذات قدرة عالية وتتناسب مع بيئة برنامج نظام ويندوز Windows

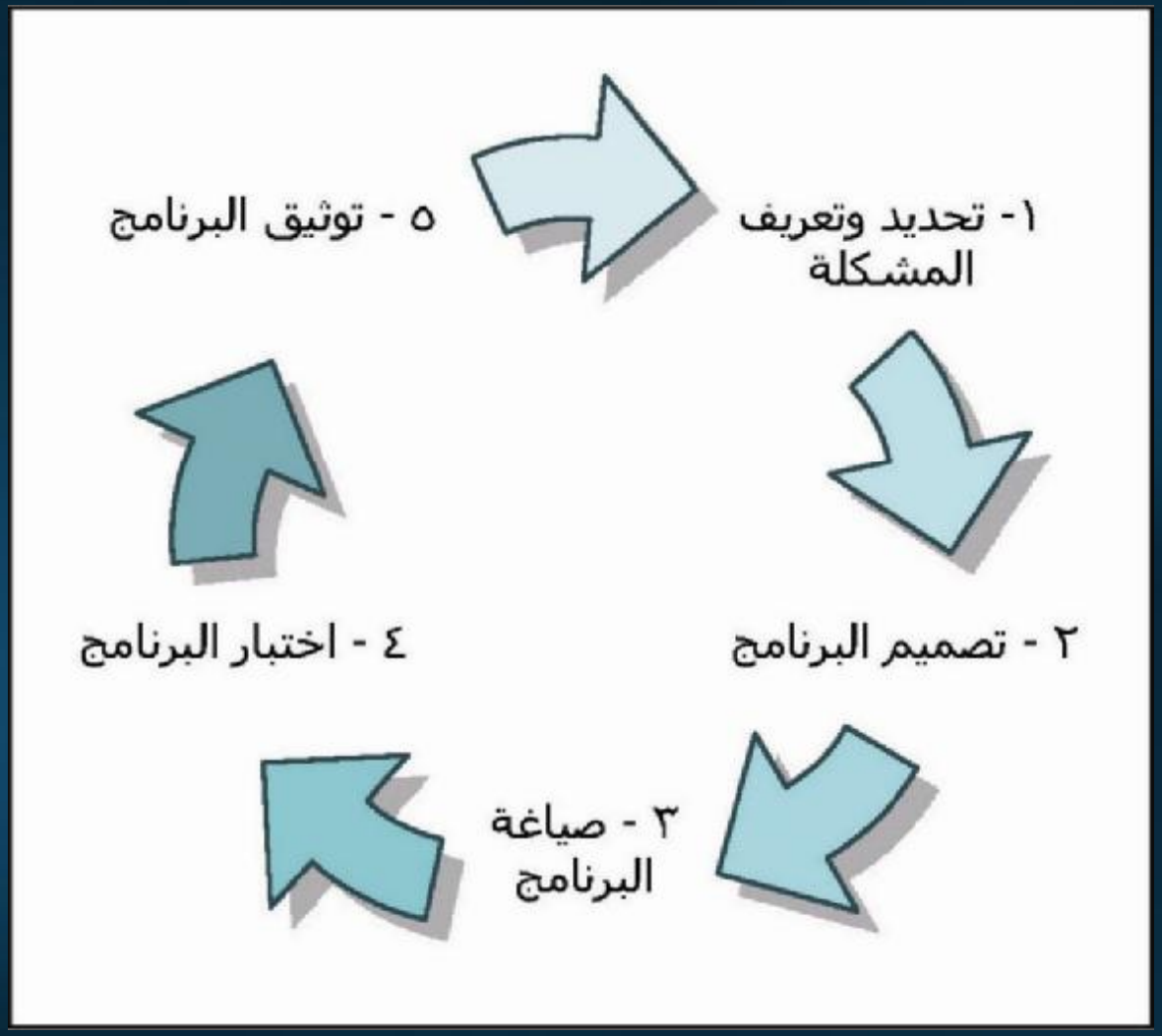
٢- لغة سي ولغة سي C & C++ Language .

- تتميز هذه اللغة بالقوة والمرونة والقدرة على إنتاج برمجيات متعددة وذات كفاءة عالية. وقد ظهرت نسخة حديثة من لغة C ذات بيئة مرئية وهي لغة C++ تتميز بكونها لغة برمجة مرئية Visual .

٣. لغة الجافا Java Language

تعتبر لغة الجافا من اللغات عالية المستوى وتعرف بأنها من اللغات المرئية Visual والشيئية Objects. تشبه لغة الجافا لغة C++ إلا أنها تتسم بالسهولة. لغة الجافا من اللغات العامة الأغراض والتي تستخدم لإنتاج برمجيات متنوعة.

خطوات صياغة وتطوير البرامج Program Development Steps



تحديد المشكلة

١ - تحديد وتعريف المشكلة Defining the Problem

في هذه الخطوة يقوم المبرمج بتحديد وتعريف المشكلة وتتضمن هذه الخطوة تحديد التالي بالترتيب:

أ - الهدف من البرنامج: كأن يكون على سبيل المثال لحساب صافي الربح، أو الرواتب أو فواتير استهلاك الكهرباء والماء، أو حساب المعدل التراكمي للطلاب.

ب - نوع وحجم المخرجات ووسائل الإخراج (تقارير - فواتير - شيكات - نقود...).

جـ - نوع وحجم البيانات المدخلة ووسائل الإدخال.

د - مستخدمي البرنامج والمستفيدين منه.

مثال

- اوجد قيمة المتغير الناتج من المعادله:

Z=(x-y)**2

خطوات الحل:

١-فهم المسأله: هو حساب قيمة المتغير z المعطى بالمعادله السابقه و تحديد المدخلات و هى x و y و هو ايجاد قيمة المتغير z المحدد بالمعادله السابقه.

٢- مرحلة التحليل: استعراض الطرق المختلفه للحل و اختيار انسبها من ناحية السرعه و السهوله و كذلك الدقه.

يوجد طريقتان للحل :

الطريقه الاولى :

١-التعويض بقيمة كل من المتغيرين x ، y

٢-ايجاد ناتج x-y

٣- ايجاد مربع الناتج السابق للحصول على قيمة المتغير z

الطريقه الثانيه :

حساب قيمة المتغير z من خلال المعادله

Z=x**2-y**2-2*x*y

١-التعويض بقيمة كل من المتغيرين y and x

٢-ايجاد مربع المتغير x=x**2

٣-ايجاد مربع المتغير y=y**2

٤-ايجاد قيمة 2*x*y

٥- ايجاد مجموع مربعى المتغييرين x and y = x**2+y**2

٦- ايجاد قيمة المتغير z من خلال طرح ناتج الخطوه 4 من تاتج الخطوه ٥

بتحليل الطريقتين السابقتين يتضح ان الاولى اسرع و اسهل فى الوصول الى الحل

تمرين

حلل المشكله الرياضيه الاتيه

y =sqr(+5)

اوجد الطرق الرياضيه المختلفه للحل و اختار الانسب

تصميم البرامج

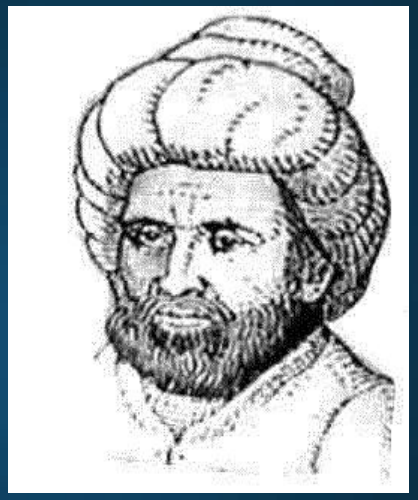
٢ - تصميم البرنامج Design the Program

يقصد بتصميم البرنامج تحديد المواصفات والخطوات الدقيقة والمرتبة منطقياً والتي تم فهمها ودراستها في الخطوة الأولى ويتم ذلك باستخدام عدة طرق منها:

الخوارزميات

ما المقصود بالخوارزميات ؟

- الخوارزمية هي عبارة عن الخطوا ت اللازمة لحل مسألة ما، و قد تكتب هذه الخوارزمية باللغة العربية أو الإنجليزية أو قد يعبر عنها برسم أشكال هندسية معينة.
- و سُميت الخوارزمية بهذا الاسم نسبة إلى العالم المسلم أبو جعفر محمد بن موسى الخوارزمي الذي ابتكرها في القرن التاسع الميلاد ي.
- فالخوارزمية تتضمن عنصريين هامين:
  - مجموعة من الخطوا ت.
  - ترتيب معين.



في هذه المحاضرة

- ما هي الخوارزميات ؟
- لماذا نحتاجها ؟
- طرق تمثيلها
  - Pseudo code
  - Flow Chart
- قواعد كتابتها
- أمثلة و تمارين عملية عليها

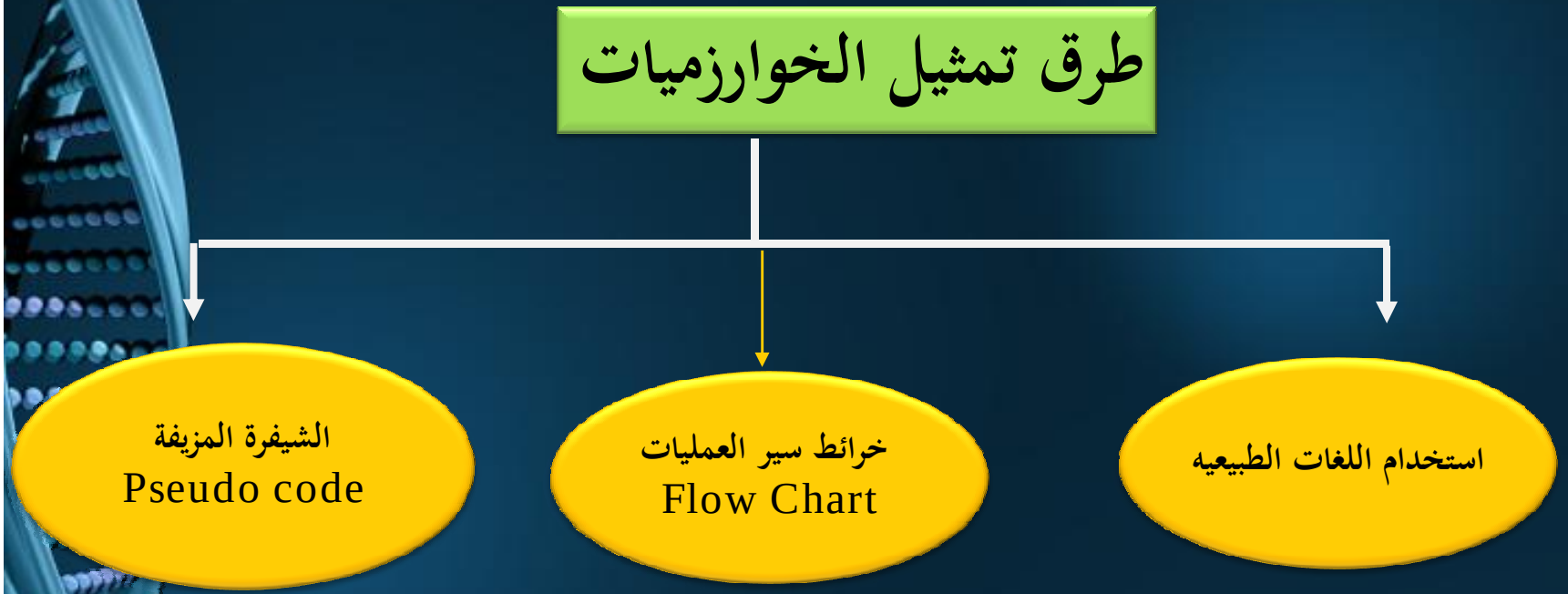
ما هي الخوارزميات ؟

- الخوارزمية يجب أ ن تتكون من تراكيب ثلاثة فقط (سويًا، فرلد ر) :
  - التسلسل .
  - الاختيار .
  - التكرار.
- و هذه التراكيب تم استنباطها من واقع أ ي عملية في حياتنا ... تمعن معي التراكيب مرة أخرى.
- للقيام بفرش قطعة موكيت في أحد الساحات ، تحتاج للقيام بمجموعة من الخطوا ت المتسلسلة المتتالية . (التسلسل)
- لإخبار طالب بأنه ناجح أو راسب تحتاج إلى مقارنة درجته بدرجة النجاح ثم تختار له النتيجة من ناجح أو راسب. (الاختيار)
- لملأ غرف البيت بالكراسي تحتاج إلى تكرار عملية الدخول للغرف الواحدة تلو الأخرى و تكرار عملية صف الكراسي. (التكرار)

لماذا نحتاجها ؟

- توثيق التفكير من أجل حل المشاكل البرمجية.
- تحديد الوقت الذي يحتاجه الكمبيوتر لحل المشكلة.
- تحديد المساحة التخزينية التي قد يحتاجها الكمبيوتر لحل المشكلة.
- المفاضلة بين الطرق و الخوارزميات من حيث السرعة و المساحة التخزينية.
- تساهم في سرعة اكتشاف أخطاء التفكير قبل البدء في مرحلة التطبيق العملي
- تساهم في إيصال فكره حل المسألة بعيدا عن الأكواد الخاصة بلغات البرمجة المتعددة
- تعطينا الفرصة لحل المشاكل بطرق مختلفة
- تساهم في سهولة فهم الأفكار و طرق الحل للمشاكل المختلفة

طرق تمثيلها



واجب

في صفحة واحدة اكتب تقرير حول تعريف و استخدامات و أهمية كلا من طرق التمثيل المذكورة أعلاه من دون المذكور في هذه المحاضرة

استخدام اللغات الطبيعه

طريقه مباشره للتعبير عن الخوارزم و ذلك بتوضيح خطوات الحل بواسطه جمل و عبارات اللغات الطبيعيه كالعربيه و الانجليزيه و....

مثال: اوجد الخوارزم لخطوات عملك اليومى

١- الاستيقاظ الساعه الثامنه

٢-االصلاه و ارتداء الملابس

٣- الوصول الى العمل احيانا بالعربيه و بواسطة التاكسى

٤-مباشره مهام العمل من مكاتبات و محاضرات الى آخره

٥- الاستعداد لمغادرة العمل الساعه الخامسه و احيانا خمسه و نصف

٦-الوصول الى المنزل تعبان و مرهق

تمرين:

اذكر الخوارزم لخطوات وصولك للجامعه

عيوبها:

١- طولها احيانا

٢- عدم الدقه

٣- اختلافها من شخص الى اخر فى طريقة صياغتها

الحل:

استخدام طرق بديله مثل"

كود الشفره (الشفره الزائفه) Pseudo code

المخططات الانسيابيه Flow charts

Pseudo code

- طريقة ذكية لتمثيل البرامج(الخوارزميات).
- شبيهة بلغة الإنسان.
- لا تعتبر لغة برمجة.
- يمكن تحويلها بسهولة للغات البرمجة المختلفة مثل جافا ، سي شار ب ، ....
- قد تكتب بأي لغة من لغات الإنسان.

```
If student's grade is greater than or equal to 60
        Print "passed"
else
        Print "failed"
```

إذا كان الطالب معدله التراكمي أكبر من أو يساوي 60
اطبع ناجح
و إلا
اطبع راسب

قواعد كتابة الـ Pseudo code

- لا يوجد في علم الخوارزميات قواعد ثابتة لتمثيل الخوارزمية بهذه الطريقة، لكن يوجد بعض الضوابط التي يجب أخذها بعين الاعتبار أثناء التمثيل، و هي:
  - لا يهم استخدام أ ي نوع من لغات البشر (العربية ، الإنجليزية ، الفرنسية، ...) .
  - يفضل أ ن تكون الكلمات المستخدمة سهلة قدر الإمكان و الأسلوب واضح.
  - يتم كتابتها ضمن مفهوم (متسلسل ، اختيار ، تكرار).
  - يتم تنظيمها ضمن نطاق منظمة و مفصلة.
  - أظهر أثناء الكتابة الخطوا ت الضمنية (التابعة) و المتكررة و المتسلسلة.
  - ابتعد عن استخدام كلمات لها معنى مقتصر على لغة برمجة معينة.
  - رتب الخطوا ت ترتيبا صحيحا

مثال11 لكتابة الـ Pseudo code

- مستخدمة الشيفرة المزيفة، اكتبي الخوارزمية المناسبة إذا أردنا أ ن نطبع متوسط درجات الحرارة للقراءا ت الثلاثة التالية T1, T2, T3.

الحل:

طريقـة ١

اقرأ قيمة درجات الحرارة T1, T2, T3
استخدم المعادلة: المتوسط = (T1 + T2 + T3) \ 3
احسب متوسط درجات الحرارة
اطبع المتوسط

طريقـة ٢

اقرأ قيمة درجات الحرارة T1, T2, T3
اجمع قيم الدرجات الثلاثة في مجموع
احسب متوسط درجات الحرارة بقسمة المجموع على ثلاثة
اطبع المتوسط

مثال2لكتابة الـ Pseudo code

- مستخدمة الشيفرة المزيفة، اكتبي الخوارزمية المناسبة إذا أردنا أن نحسب متوسط درجات الحرارة للقراءات الثلاثة التالية T1, T2, T3 مع طباعتها إذا كانت أكبر من 15درجة و إلا اطبعي الجو بارد.

الحل:

اقرأ قيمة درجات الحرارة T1, T2, T3
استخدم المعادلة: المتوسط = (T1 + T2 + T3) \ 3
احسب متوسط درجات الحرارة
إذا كان المتوسط أكبر 15درجة
اطبع المتوسط
و إلا
اطبع جملة (الجو بارد)

واجب

أعيديتمثيل الخوارزمية المطلوبه أعلاه مع طباعة (الجو بارد إن كانت الحرارة أقل من ١٥) و طباعة (الجو معتدل إن كانت بين ١٥ و ٢٥) و طباعة حر فيما غير ذلك

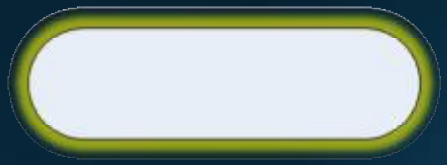
خرائط سير العمليات Flow Chart

- و هي عبارة عن طريقة لبيان خطوا ت حل المسألة وكيفية ارتباطها ببعض باستخدام رموز اصطلاحية لتوضيح خطوا ت الحل (تمثيل رمزي للخوارزمية).
- لا تحتاج إلى التعبير بلغتك كثيرًا.
- فقط تحتاج لمعرفة الرمز المناسب و الترتيب المناسب.
- لهذا يعتبرها الكثير أنها الأسهل.

قواعد التمثيل بخرائط سير العمليات

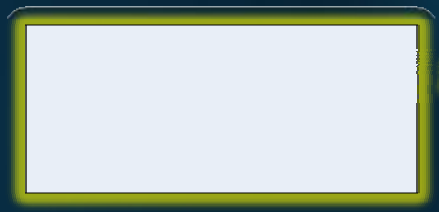
- ضوابط تمثيل الخوارزمية باستخدام خرائط سير العمليات التي يجب أخذها بعين الاعتبار أثناء التمثيل، و هي:
  - لا يهم استخدام أ ي نوع من لغات البشر (العربية ، الإنجليزية ، الفرنسية، ...) .
  - احرص على استخدام الشكل المناسب للخطوة المناسبة.
  - يفضل أ ن يكون الأسلوب واضح.
  - يتم كتابتها ضمن مفهوم (متسلسل ، اختيار ، تكرار).
  - أظهر أثناء الكتابة الخطوا ت المتسلسلة و المتكررة و الاختيارية.
- الرموز التي ستستخدم في هذا الموضوع مبينة في الشريحة التالية...

خرائط سير العمليات Flow Chart

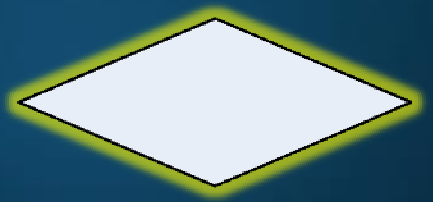


1) تستخدم لتحديد بداية أو نهاية البرنامج وتكون إحدى قيمتين Start أو Stop

2) تستخدم للتعبير عن تعليمات استقبال أو إخراج البيانات أي Input أو Output



3) تستخدم للتعبير عن تعليمات معالجة (العمليات الحسابية) أو تخزين البيانات



4) تستخدم للتعبير عن العمليات المنطقية أو عمليات التقرير

5) تحدد اتجاه سير البرنامج أي تسلسل خطوات الحل

أنواء خرائط سير العمليات

- وهناك عدة أنواء لخرائط سير العمليات تختلف هذه الأنواء حسب العناصر المكونة لها وهذه الأنواء هي:
  - خرائط التتابع البسيط.
  - خرائط التفرع.
  - خرائط الدورا ن البسيط.
  - خرائط الدورانات المتعددة أو المتداخلة.
- ويمكن أ ن يشتمل الحل على واحدة أو أكثر من هذه الأنواء الأربعة.
- خرائط التتابع البسيط: وهو نوع من الخرائط بشكل سلسلة مستقيمة من بداية البرنامج حتى نهايته بحيث تنعدم فيها التفرعات وتخلو من الدورانات.

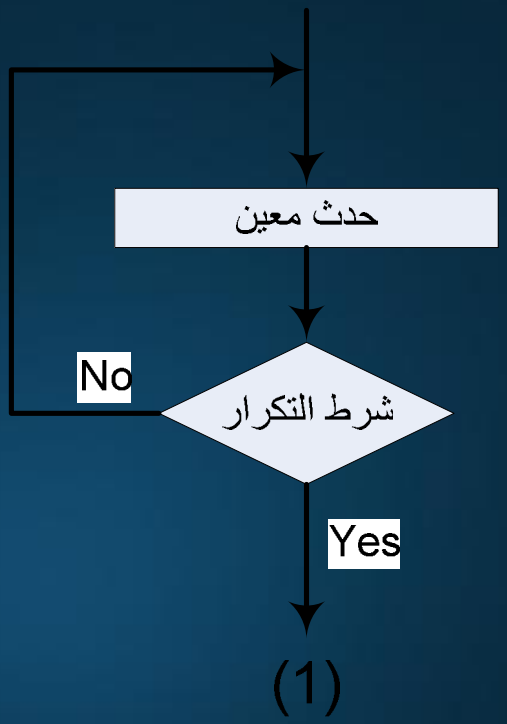
خرائط التفرع

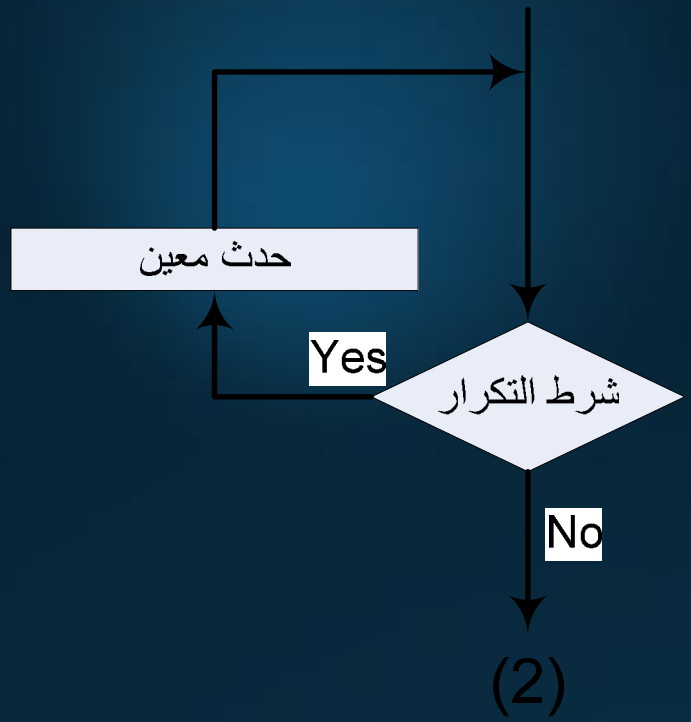
- وهو عبارة عن نوع من الخرائط يحتوي على تفرع بسبب الحاجة لاتخاذ قرار، أو المقارنة بين اختيارين أو أكثر (باستخدام العمليات المنطقية)، فيسير كل اختيار في اتجاه مختلف عن الآخر. والشكل التالي يبين هذين المسارين وهما ناتج العملية المنطقية (صح أو خطأ).

خرائط الدورا ن

- كما نعلم أ ن من أهم مزايا الحاسوب القدرة على التكرار أ ي إعادة عملية معينة من دو ن الحاجة إلى تكرار كتابتها ولكن وفقاً لشرط معين، في حال عدم تحقق هذا الشرط تتوقف عملية التكرار، وإلا استمر في التكرار إلى ما لا نهاية.
- خرائط الدورا ن هي عبارة عن خرائط تستخدم للتعبير عن تكرار أو إعادة عملية أو مجموعة من العمليات في البرنامج عدداً محدوداً أو غير محدود من المرا ت، ويكون الشكل العام لهذه الخرائط كما في الشكل التالي.

خرائط الدورا ن



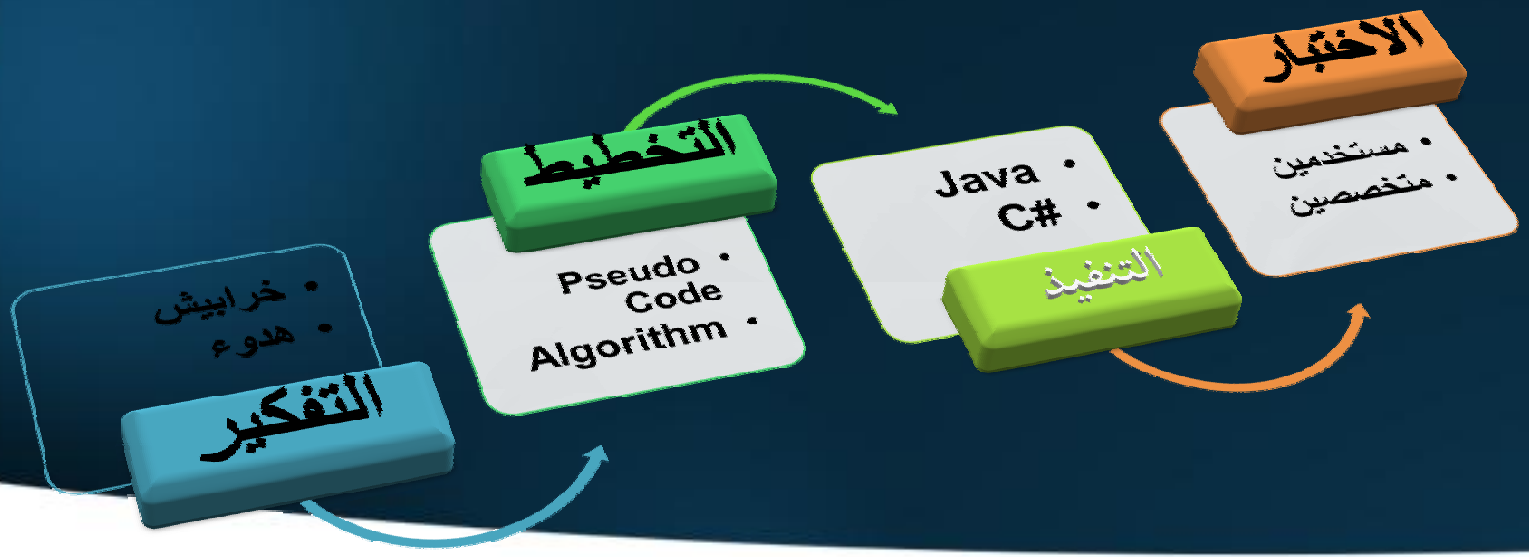


مثال لكتابة الـ Flow chart

- مستخدمة خرائط سير العمليات . اكتب الخوارزمية المناسبة إذا أردنا أ ن نحسب متوسط درجات الحرارة للقراءا ت الثلاثة التالية T1, T2, T3.

خطوا ت هامة لبناء أ ي برنامج

- صناعة أ ي شيء يحتاج إلى التفكير ... التخطيط .... التنفيذ ... الاختبار
- و لصناعة برنامج ، نقم بالتالي:
- ماذا الذي أريد بناءه؟ (خرابيـــش!!!)
- كيف أبنيه ؟ Algorithm, pseudo code
- كيف أنفذ ؟ Java, C#, C++,
- كيف أختبر ؟



خطوات هامة لبناء أي برنامج

- في كل مشكلة برمجية بعد ذلك ستقوم بعرضها على المراحل الثلاث التالية على الأقل للوصول لحل مميز و صحيح.

تمارین و تدریبات شاملة

تدريب 1.1

- نحتاج لإيجاد مساحة الدائرة التي نصف قطرها R ، علمًا بأن مساحة الدائرة تساوي π * R2 ، حيث π هي عبارة عن ثابت قيمته دائماً ٣.١٤ تقريباً.

تنبيه هام

في مثل هذا النوع من التدريبات يكون الاختيار للطالب بين تمثيل الخوارزمية بالشيفرة المزيفة أو مخطط سير العمليات.

الخوارزمية باستخدام الشيفرة المزيفة

اقرأ قيمة نصف القطر R
ضع قيمة π = 3.14
احسب المساحة من المعادلة التالية، المساحة = π * (R * R)
اطبع الناتج

تدريب 1.1

الخوارزمية باستخدام خرائط سير العمليات

تدريب 1.2

- مستخدمــة طريقة الشيفرة المزيفـة ، نحتاج لإيجاد محيط ملعب كرة القدم ، و تحديد إن كان دولي (أكبر من أو يساوي 600 م) أم محلي (أقل من ذلك)، علمًا بأن محيط المستطيل تساوي (الطول + العرض) × 2 .

الخوارزميـة باستخدام الشيفرة المزيفة

تنبيـه هام

اقرأ قيمة عرض الملعب W
اقرأ قيمة طول الملعب L
احسب المساحة من المعادلة التالية، المحيط= 2 * (L + W)
إذا كانت المساحـة أكبر من أو يساوي 600 متر مربع
اطبع المحيط
اطبع الملعب دولي
و إلا
اطبع المحيط
اطبع الملعب دولي

في هذه الحالة أنت تكوني مجبــرةً علـــى تمثيـــل الخوارزميـــة بالشـــيفرة المزيفـــة ، لأنَّ الســـائل أجبركِ على ذلك .

واجبـــــ

أعيد تمثيل الخوارزميـة المطلوبه أعلاه مع لحساب محيط الملعب مع مساحته.

تدريب 1.2

الخوارزمية باستخدام خرائط سير العمليات

تدريب 1.3

- إذا علمت أ ن قسم علوم الحاسب بالكلية يضم A تخصصات و في كل تخصص يوجد في الفصل الأو ل B مساقات و في كل مساق يوجد C شعبة، و كل شعبة تحتوي على عدد Dمن الطلبة، احسب عدد الطلبة الذين ينتمون للقسم في هذا الفصل

الخوارزمية باستخدام الشيفرة المزيفة

اقرأ قيمة عدد التخصصات A
اقرأ قيمة عدد المساقات في كل تخصص B
اقرأ قيمة عدد الشعب للمساق الواحد C
اقرأ قيمة عدد الطلبة في الشعبة الواحدة D
احسب عدد الطلبة بالمعادلة التالية: عدد الطلبة = A*B*C*D
اطبع عدد الطلبة

واجب

أعيد تمثيل الخوارزمية المطلوبه أعلاه باستخدام مخطط سير العمليات.

واجب

أعيد تمثيل الخوارزمية المطلوبه أعلاه مع اعتبار أن عدد الطلبة غير متساوي في الشعب

تدريب 1.4

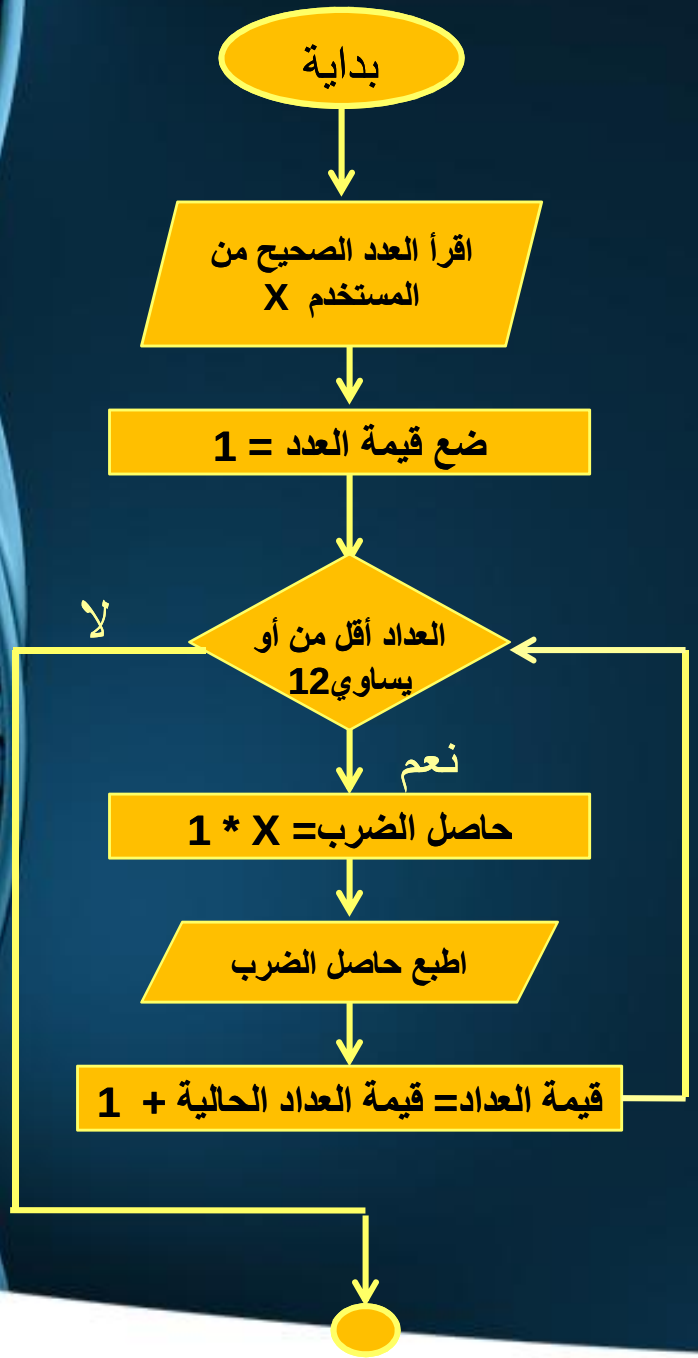
- نحتاج لانجاز برنامج يستقبل من المستخدم رقم صحيح ثم يقوم بطباعة حاصل مضروباته من ١ حتى ١٢.

الخوارزمية باستخدام الشيفرة المزيفة

اقرأ قيمة العدد الصحيح من المستخدم X
حدد رقم عدَّاد عند القيمة 1
كرر ما يلي بحيث قيمة العداد تكون أقل من أو يساوي 12
حاصل الضرب = X*1
اطبع حاصل الضرب
قم بزيادة العداد بواحد
اطبع (انتهيت من طباعة مضروبات العدد X)

تدريب 1.4

الخوارزمية باستخدام مخطط سير العمليات



تدريب 1.5

- اكتب الخوارزمية اللازمة لبرنامج يقوم بإيجاد قيمة الدالة F(x) المعرفة كالتالي:

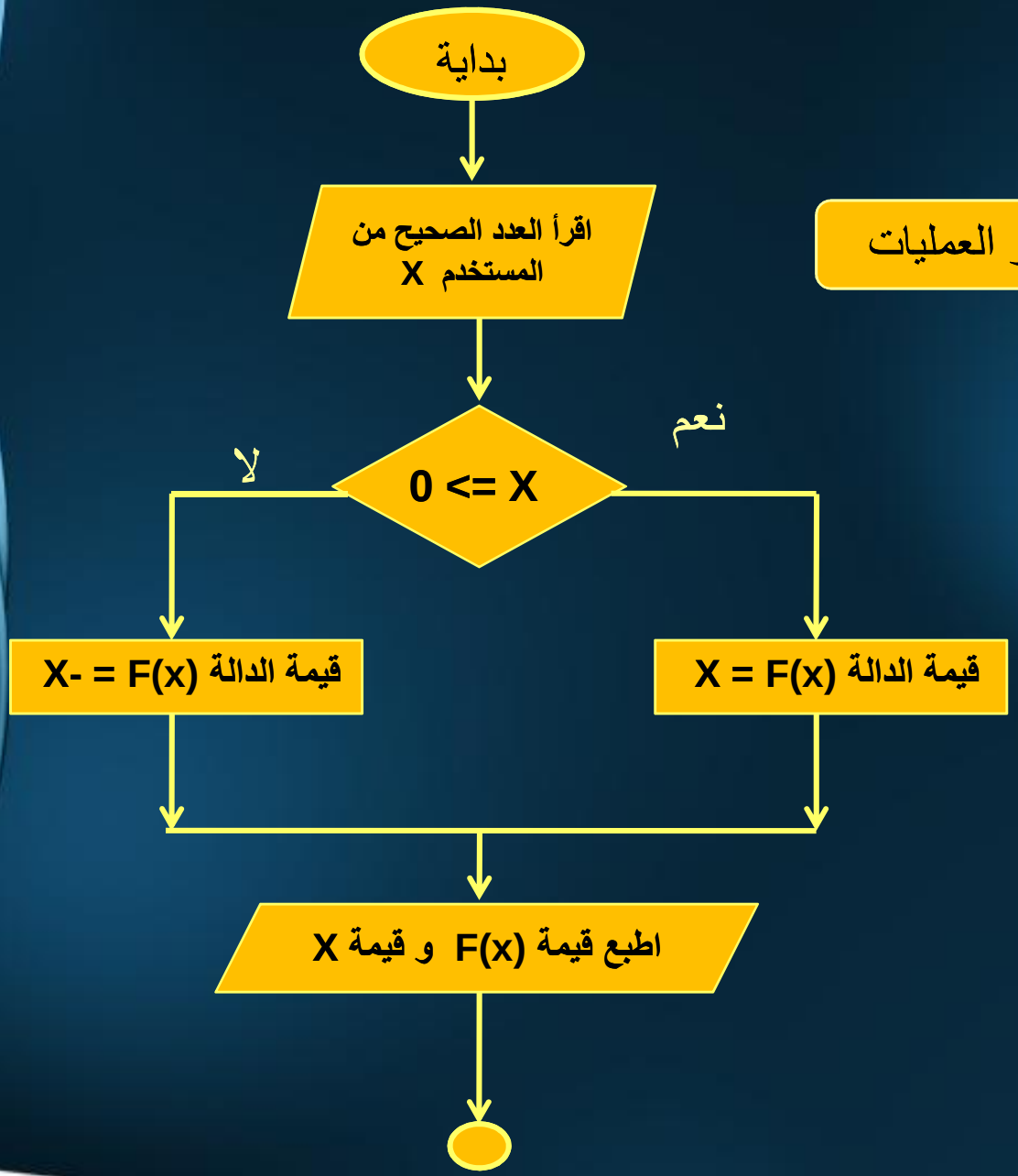
$$F(x)=\begin{cases} x & x\geq 0 \\ -x & x<0 \end{cases}$$

الخوارزمية باستخدام الشيفرة المزيفة

اقرأ قيمة المتغير x
إذا كانت x أكبر من أو تساو ي صفر
قيمة الدالة F(x) تساو ي x
وإلا
قيمة الدالة F(x) تساو ي -x
اطبع قيمة كل من x, F(x).

تدريب 1.5

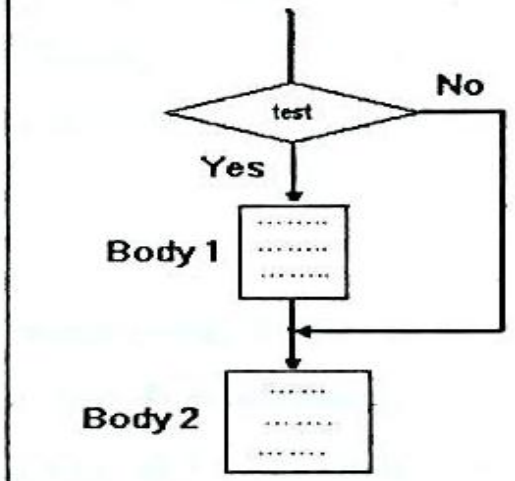
الخوارزمية باستخدام مخطط سير العمليات



If structure

الشكل رقم 1 يوضح صيغة بناء المخطط الانسيابي (Flowchart) وكود الشفرة (Pseudo code) للتركيب (IF structures)

Code	Flowchart

If (test) then
body 1
Set of instructions
(or statements)
end if
body 2
	

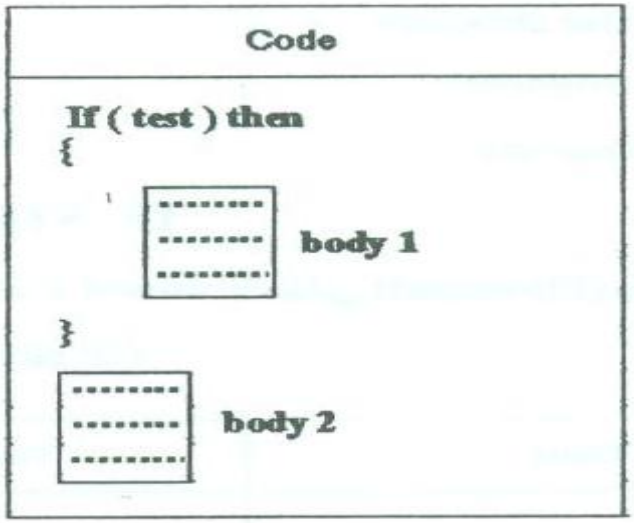
شكل 1 : صيغة بناء المخطط الانسيابي (Flowchart)
وكود الشفرة (Pseudo code) للتركيب (IF structure)

في هذا التركيب:

إذا كان الشرط (test) صحيحا يتم تنفيذ العبارات الموجودة في (body1) ثم تنفيذ العبارات الموجودة في (body2). إذا كان الشرط غير صحيحا فسيتم تنفيذ العبارات الموجودة في (body 2) دون تنفيذ العبارات الموجودة في (body 1). يتم تنفيذ العبارات الموجودة في (body 2) دائما

ملحوظة

يمكن استخدام الصيغة الموضحة في شكل 2 بناء التركيب (If structure)



شكل 2: صيغة أخرى لبناء المخطط الانسيابي (Flowchart)

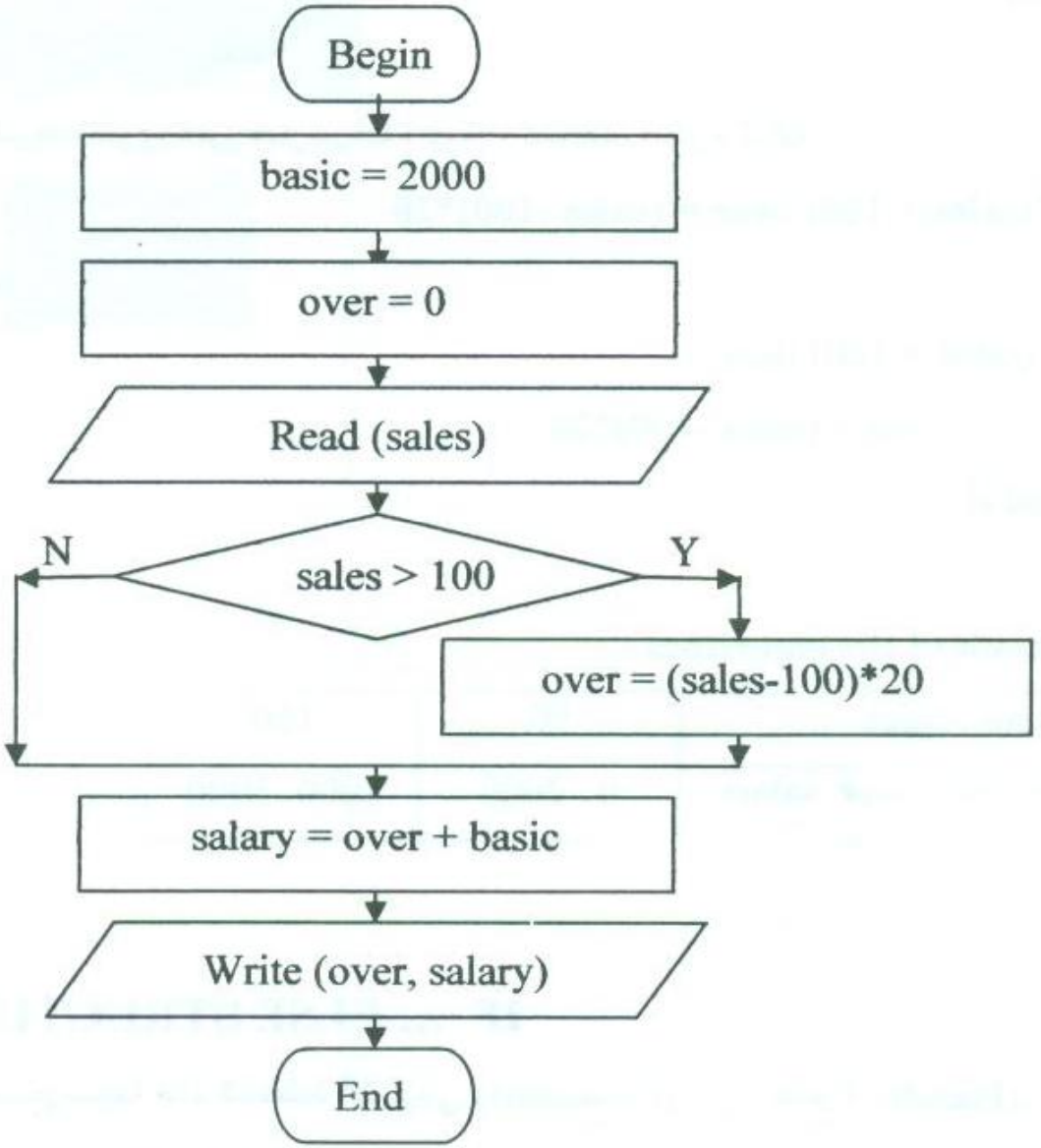
وكود الشفرة (Pseudo code) للتركيب (IF structure)

مثال

شركة لبيع لعب الأطفال تقوم بحساب راتب العامل (salary) على أساس مبيعاته (sales) من لعب الأطفال. إذا زادت مبيعات العامل عن 100 لعبه يحسب له أجر إضافي (over) 20 ريال عن كل لعبه إضافية. أوجد المخطط الانسيابي (flowchart) بالإضافة إلى كود الشفرة (Pseudo code) لخوارزم يقرأ عدد اللعب المباعة (sales) بواسطة العامل ثم يحسب ويطبع الأجر الإضافي (over) والراتب الإجمالي (salary) للعامل إذا علمت أن راتب العامل الأصلي (basic) 2000 ريال .

الحل

Flowchat



Pseudo code:

```
Variable salary, over=0, basic=2000, sales: integer
Begin
Read (sales)
If (sales > 100) then
        over = (sales -100)*20
End if
```

```
salary= over + basic
Write (over, salary)
End
```

حيث أن جسم الشرط يحتوي على عبارة واحدة فيمكننا استخدام العبارة التالية

If (sales > 100) over = (sales -100)*20

بدلا من العبارات

```
If (sales > 100) then
        over = (sales -100)*20
End if
```

Result of the algorithm

Input : sales	90	150
Output : over, salary	0 2000	1000 3000

مكونات لغة البرمجه

هي مجموعة من القواعد والمصطلحات يستخدمها المبرمج لكتابة برنامج معين يفهمه الحاسب حسب نوعه ويتم تحويل البرنامج طبقاً للغة المكتوب بها إلي لغة الآلة (machine language) بواسطة المعالجات compilers أو المجمعات assemblers

1.1.5) العناصر الأساسية للغة البرمجة

The Main Element of Software Languages

تتكون لغات البرمجة من العناصر التالية:

a) مجموعة من الرموز الأساسية

1) حروف لاتينية : A, B, C, ..., X, Y, Z
2) أرقام عربية : 0,1,2,3,4,..... 9
3) رموز خاصة : $, % , / , * , - , +

b) الكلمات

وهي نوعان:

1) **كلمات محجوزة (reserved words):** هي كلمات لها معني خاص بالحاسب وتقوم بتنفيذ مهام محددة ويوجد قيود عليها حسب لغة المبرمجة المستخدمة مثل:

Read, Write, Add, For, and If

2) **كلمات يختارها المستخدم أو المبرمج (user defined words) :** هي كلمات يختارها المبرمج أو المستخدم ولا يوجد قيود على استخدامها مثل :

Alpha, A, ABD, sum, etc.

c) مجموعة التعليمات Instructions set

هي مجموعة من الرموز أو الكلمات الخاصة بلغة البرمجة لتنفيذ عملية أو مهام أو أمر معين وتنقسم إلى ثلاثة أنواع:

1) تعليمات خاصة بالذاكرة (memory reference instructions)

وتقوم غالبا بتنفيذ العمليات الحسابية أو المنطقية مثل: ADD, *, /, +, SUB, etc. فعلى سبيل المثال العبارة

$$C = A + B$$

تعنى جمع محتوى الذاكرة الموجودة في العنوان A مع محتوى الذاكرة الموجود في العنوان B ووضع الناتج في العنوان C داخل الذاكرة

2) تعليمات إدخال وإخراج (Input/ output instructions)

هي تعليمات لإدخال البيانات وإخراج المعلومات مثل: Write and Read. فعلى سبيل المثال:

Read (X, Y, Z)

Write (A, B, C)

العبارة الأولى تعني إدخال قيم المتغيرات (X, Y and Z) من خلال وحدة الإدخال والعبارة الثانية تعني إخراج قيم المتغيرات (A, B and C) إلى وحدة الإخراج

3) تعليمات تحكم (Control instructions):

تشمل التعليمات التي تؤدي إلي تسلسل تنفيذ البرنامج مثل

Goto, While, Do-while, switch-case, for ...etc.

عندما يفهم الحاسب هذه التعليمات تقوم وحده التحكم بالجهاز (control unit) بإرسال الإشارات أو أوامر التحكم اللازمة إلي الوحدات المادية المعنية بذلك لتنفيذ هذه التعليمات

يتم تنفيذ كل تعليمة من خلال ثلاث مراحل أساسية وهى:

1) مرحلة استحضار العبارة التي تحتوى على التعليمة من ذاكرة الحاسب (Fetch)
2) مرحلة ترجمة هذه التعليمة لفهم مدلولها (decoding)
3) مرحلة تنفيذ هذه التعليمة (execution)

قواعد اللغة

d) قواعد اللغة:

هي مجموعة من القواعد والقيود التي يجب الالتزام بها عند كتابه البرامج وعند مخالفه هذه القواعد والقيود يعطي البرنامج أثناء مرحله الترجمة أخطاء لغوية (Syntax errors)

البيانات والمعلومات والتعليمات والعمليات

DATA, INFORMATION, INSTRUCTIONS AND OPERATION

1.2.1) البيانات Data

هي مدخلات الحاسب وتعتبر هي المادة الخام للبرنامج والتي لا تعطي معني بذاتها

أنواع البيانات:

1) بيانات عدديه (Numeric data):

هي الأرقام الصحيحة (integer) والحقيقية (real) والثنائية (binary) مثل:

$(123)_{10}$, $(12.56)_{10}$, and $(1011011)_2$

2) بيانات حرفيه (Character data):

هي الحروف اللاتينية مثل:

'A','B','C', ...'a',' b', 'c'.....,'X','Y'

3) كلمات (String data):

هي مجموعه من الحروف والتي تشكل كلمات مثل :

"abck", "Ahmed"

4) بيانات بولين (Boolean data) :

هي بيانات تأخذ احد القيمتين (False or True)

5) بيانات مركبة (Composite data):

تنقسم البيانات المركبة إلى نوعين رئيسيين هما المصفوفات (arrays) والسجلات /التركيبات (Records/structures)

a) المصفوفات (array):

هي مجموعة من البيانات من نفس النوع

مثل : مصفوفة الأرقام الصحيحة {12, 15, 34} التي تحتوي على ثلاثة عناصر من البيانات الصحيحة و مصفوفة الأرقام الحقيقية {1.6, 12.4, 12.6} التي تحتوي على ثلاثة عناصر من البيانات الحقيقية

b) السجلات (records) أو التركيبات (structures) :

هي مجموعة من البيانات المختلفة في النوع مثل سجلات الموظفين حيث أن كل سجل يحتوي على اسم الموظف (كلمات string) ورقم الموظف (رقم صحيح integer) وراتبه الشهري (رقم حقيقي real). فعلى سبيل المثال

("Ahemd", 12450, 'B', 2020.5)

البيانات السابقة تكون سجل أو تركيب يحتوي على أربعة عناصر. العنصر الأول "Ahmed" (string data) يمثل اسم الموظف والعنصر الثاني 12450 (integer data) يمثل رقم الموظف والعنصر الثالث 'B' (character data) يمثل درجة الموظف والعنصر الرابع 2020.5 (real data) يمثل راتب الموظف الشهري

1.2.2) التعليمات Instructions

هي مجموعه من التعليمات والأوامر تتعلق بتعليمات خاصة بالذاكرة أو تعليمات خاصة بوحدات الإدخال والإخراج وتعليمات خاصة بالتحكم وهو ما تم تناوله سابقا

1.2.3) المعلومات Information

هي البيانات التي تمت معالجتها (Processed data) من خلال إجراء عمليات وتعليمات معينه عليها لتعطي قيم لها معني بذاتها

العمليات OPERATION

يوجد ثلاث عمليات رئيسية يتم استخدامها في البرمجة وهي:

a) عمليات حسابية Arithmetic operation
b) عمليات منطقية Logical operations
c) عمليات العلاقة Relational operations

a) العمليات الحسابية Arithmetic operations

الجدول رقم 1 يوضح العمليات الحسابية وعامل (operator) كل عملية

جدول 1: العمليات الحسابية وعامل (operator) كل عملية

العملية	العامل operator
الجمع	+
الضرب	*
الطرح	-
القسمة	/
باقي القسمة	% أو MOD
الأس	** أو ^

b) العمليات المنطقية Logical operations

الجدول رقم 3 يوضح العمليات المنطقية وعامل (operator) كل عملية. الجدول رقم 4 يوضح ناتج العمليات المنطقية

جدول 3: العمليات المنطقية وعامل (operator) كل عملية

العملية	العامل operator
NOT	!
AND	&&
OR	||

جدول 4 : ناتج العمليات المنطقية

A	B	A&&B	A||B	!A	!B
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

المعاملات (operands) تكون بيانات ثنائية (0 or 1). ناتج العملية يكون بيانات ثنائية (0 or 1). فعلى سبيل المثال إذا كان $A = (1)_2$, and $B = (0)_2$ فان:

$C1 = A \&\& B \longrightarrow C1 = (0)_2$

$C2 = A \| B \longrightarrow C2 = 1$

$C3 = !A \longrightarrow C3 = 0$

حيث أن A, B, C1, c2, and C3 بيانات ثنائية

c) عمليات العلاقة **Relational operations**

الجدول رقم 5 يوضح عمليات العلاقة وعامل (operator) كل عملية

المعاملات (operands) تكون بيانات عددية: صحيحة (integer) أو حقيقية (real) أو ثنائية (binary) . ناتج عمليات العلاقة يكون بيانات من النوع Boolean (False or True) . فعلى سبيل المثال إذا كان A = 10 and B = 5 فان :

C1 = A > B ⟶ C1 = True (T)

C2 = A < B ⟶ C2 = False (F)

C3 = A ==B ⟶ C3 = False (F)

حيث أن A, and B بيانات تحتوى على أرقام صحيحة. C1, C2, and C3 بيانات Boolean تأخذ قيم False (F) or True (T). إذا تحقق الشرط يكون ناتج العلاقة يساوي True (T) وإذا لم يتحقق الشرط يكون ناتج العلاقة False (F).

تستخدم عمليات العلاقة في التركيبات الشرطية وهو ما سنتناوله في الفصل الثالث

جدول 5: عمليات العلاقة وعامل (operator) كل عملية

العملية	العامل operator
يساوي	==
لا يساوي	!=
اقل من	>
اقل من أو يساوي	>=
اكبر من	<
اكبر من أو يساوي	<=

1.3) المتغيرات والثوابت VARIABLES AND CONSTANTS

البيانات والمعلومات التي نتعامل بها من خلال البرامج لها صنفان هما المتغيرات (Variables) والثوابت (Constants)

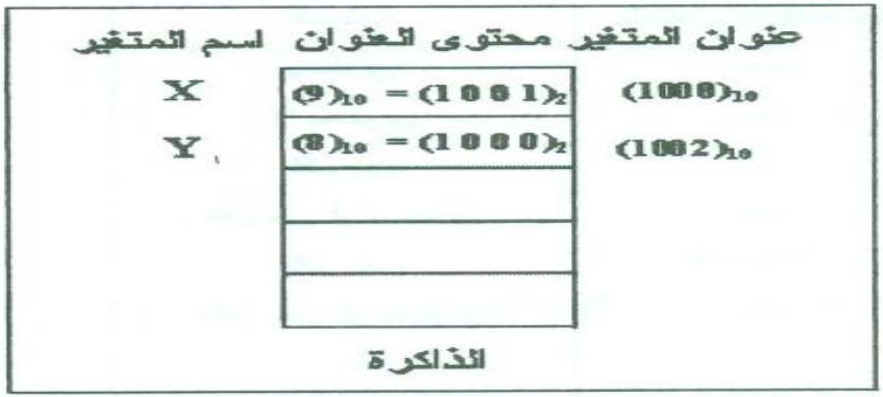
1.3.1) المتغيرات Variables

هي بيانات تتغير قيمتها أثناء تنفيذ البرنامج. يمكن تسمية هذه البيانات بحروف أو كلمات لتمثل عنوان (موضع) تخزين قيمة البيانات (المحتوي) في الذاكرة حيث أن المحتوي يتغير أثناء تنفيذ البرنامج ويمكن القول أن محتوي العنوان يمثل بيانات أو معلومات يمكن أن تتغير أثناء تنفيذ البرنامج. فعلى سبيل المثال

$X = (9)_{10} = (1\ 0\ 0\ 1)_2$

$Y = (8)_{10} = (1\ 0\ 0\ 0)_2$

المتغير X يمثل بالعنوان $(1000)_{10}$ في الذاكرة والذي يحتوي على القيمة $(9)_{10}$ في النظام العشري أو القيمة $(1001)_2$ في النظام الثنائي. المتغير Y يمثل بالعنوان $(1002)_{10}$ في الذاكرة والذي يحتوي على القيمة $(8)_{10}$ في النظام العشري أو القيمة $(1000)_2$ في النظام الثنائي. الشكل رقم 3 يمثل هيكلا لجزء من ذاكرة الحاسب يتضمن محتوى وعنوان المتغيرين X and Y



شكل 3: هيكلا لجزء من ذاكرة الحاسب يتضمن محتوى وعنوان المتغيرين X and Y

عند الإعلان عن المتغيرات يجب تحديد نوع البيانات (Data) الذي يحتويها هذا المتغير

أمثلة:

Variable X, Y: integer

الإعلان عن المتغيرين X, and Y من النوع integer

Variable A1, B2: real

الإعلان عن المتغيرين A1, and B2 من النوع real

Variable ch: character

الإعلان عن المتغير ch من النوع character

1.3.2) الثوابت Constants

هي بيانات لا تتغير قيمتها أثناء تنفيذ البرنامج

أمثلة:

Constant A3=20: integer

A3 مقدار ثابت من النوع integer قيمته 20

Constant AA = 12.4 real

AA مقدار ثابت من النوع real قيمته 12.5

Constant name = "Ali": string,

Or

Constant name[5] = "Ali": character

name مقدار ثابت من النوع string قيمته "Ali"

عند وجود الثابت داخل جسم البرنامج يقوم المترجم (compiler) بالتعويض عن قيمة الثابت بالقيمة المحددة عند الإعلان عنه. فعلى سبيل المثال في العبارة

Y = A3 + 10

يقوم المترجم بالتعويض عن الثابت A3 بالقيمة 20 المعرفة مسبقا في العبارة

Constant A3 = 20 : integer

وبالتالي تكون قيمة المتغير Y تساوي 30

1.4) قواعد عامة متعلقة بأنواع البيانات

GENERAL RULES FOR DATA TYPES

1) تعريف البيانات قبل استخدامها:

قبل استخدام أي بيانات في البرنامج يجب تعريفها. يتم تعريف البيانات من ناحيتين

a) ناحية الغرض :Data object

- Variable
- Constant

b) ناحية النوع Data type

- Integer
- Real
- Character
- String
- Boolean

2) في معظم لغات البرمجة لا يصح إجراء عمليات علي بيانات من أنواع مختلفة (different data types) فعلى سبيل المثال إذا تم الإعلان عن المتغيرات Y, F, X and K بالعبارات التالية

Variable Y,F: integer
Variable X: character
Variable K: real

واستخدمت العبارة التالية

Y = X + F

فإن العباره خطأ

1.5) الأسبقيات PERIORITIES

في الجزء التالي سنناقش أسبقيات تنفيذ العمليات الحسابية والمنطقية والمختلطة

1.5.1) أسبقيات العمليات الحسابية Priorities of Arithmetic Operators

لتنفيذ العمليات الحسابية يجب إتباع الترتيب التالي:

1) الأقواس
2) الأس
3) الضرب والقسمة وباقي القسمة (من اليسار إلي اليمين)
4) الجمع والطرح (من اليسار إلي اليمين)

أجري تنفيذ المعادلات التالية

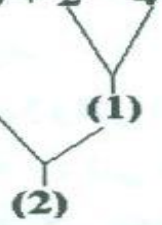
a) X = 3 + 2 * 4

b) Y = 2 * 6 + 3 – 2 * 4

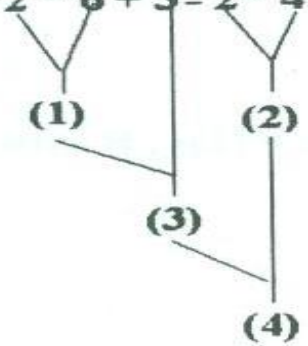
c) Z = (2*5**2)/ 2+12

الحل:

a) x = 3 + 2 * 4 = 3 + 8 = 11



b) y = 2 * 6 + 3 – 2 * 4 = 12 + 3 – 2 * 4 = 12 + 3 – 8 = 15 - 8 = 7



c) Z = (2*5**2)/ 2+12

= (2*25)/2+12

= (50)/2+12

= 25 +12 =27

مثال: 3

1.5.2) أسبقيات العمليات المنطقية Priorities Of Logical Operations

لتنفيذ العمليات المنطقية يجب إتباع الترتيب التالي:

1) الأقواس
2) NOT والتي تكافئ !
3) AND والتي تكافئ &&
4) OR والتي تكافئ ||

مثال: 4

إذا كان $A = (1)_2$, $B =(0)_2$ أوجد قيمة Z من المعادلات التالية :

a) Z = !A && B || A

b) Z = ! (A || B) && B

الحل:

a) Z = !A && B || A

=!0 && 0 || 1

= 1 && 0 || 1

= 0 || 1 = 1

b) Z = ! (A || B) && B

= ! (1 || 0) && 0

= ! (1) && 0

= 0 && 0 = 0

1.5.3) أسبقيات العمليات المختلطة Priorities Of Combined Operations

في معظم المسائل يكون لدينا عمليات مختلطة. الجدول رقم 6 يوضح ترتيبا لأسبقيات تنفيذ العمليات المختلطة (عمليات حسابية ومنطقية وعلاقة):

جدول 6: ترتيب أسبقيات تنفيذ العمليات المختلطة

الترتيب		Operators
1	الأقواس: ينفذ ما بداخل القوس	( )
2	العمليات أحاديه المتغير : Single operand Unary: Logical Not : !a arithmetic minus : -a	(!) (-)
3	arithmetic operators حسب ترتيب العمليات الحسابية	(**) (*) (/) (Mod)
4	arithmetic operators تنفذ من اليسار إلي اليمين	(+) (-)
5	relational operators تنفذ من اليسار إلي اليمين	(>) (<) (<=) (>=)
6	relational operators تنفذ من اليسار إلي اليمين	(==) (!=)
7	Logical AND, OR حسب ترتيب العمليات المنطقية	(&&) (||)
8	Assignment operators عوامل التوظيف	(=)(+=)(-=)(*=) (/=) (%=)

مثال: 5

إذا كان A = 1 0 أوجد قيمة Z من المعادلات التالية:

Z = (3 > 2) && A || (2 < 6)

الحل:

Z = (3 > 2) && A || (2 < 6)

= T && T || T

= T || T = T

مثال: 6

إذا كان C = 4, and D=2 . أوجد قيمة Y من المعادلة التالية

Y = (C **D > 8) || (A && D = 4)

الحل:

Y = (C **D > 8) || (A && D = 4)

= (4**2 > 8) || (T && 2 =4)

= (16 > 8) || (T && F)

= T || F = T

تمارين

1) إذا كان . A = 3 , B = 7 , C=12 and D=2 أوجد قيمة العبارات التالية

a) A > 3 || B <= 5

b) !(A < 5) && (A > D)

c) (A+B) < D || !(C < D)

d) A < B + 5 && C + 5 < D

e) A** 2/5+2 >= 3 || 5 < D

f) A +B / 2 * 2 < D && !(D == A)

g) A != B && A < C || A < D

2) احسب نتيجة العمليات التالية:

- إذا كانت المتغيرات a, b, c, and d هي أعداد صحيحة (Integers) حيث أن قيمها:

a=4, b=9, c=8, and d= -1

- المتغيرات a and b هي أعداد صحيحة والمتغيرات c and d هي متغيرات ثنائية حيث أن قيمها: $a=5, b=3, c= (1)_2$, and $d=(0)_2$

a < b || !(c && b > a) && d

(a * 2 > 1) && d || !(a - b <2 && !(c))

3) إذا كان $A=(1)_2$, $B=(0)_2$ أوجد قيمة Z من المعادلات التالية :

a) **Z = ! A || B && A**

b) **Z = ! (A && B) N|| B**

c) **Z = A < B**

d) **Z = (3 > 2) N&& A || (2 < 6)**

4) إذا كان C = 7, and D=2 . أوجد قيمة Y من المعادلة التالية

Y = (C **D < 8) && (A && D = 4)