كلية العلوم

القسم : **الفيزياء**

السنة : **الثانية**

المادة : لغات البرمجة١

المحاضرة : الثالثة والرابعة/نظري/

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

2025

كلية العلوم ، كلية الصيدلة ، الهندسة التقنية

١١

يمكنكم طلب المحاضرات برسالة نصية (SMS) أو عبر(What's app-Telegram)على الرقم 0931497960

الفصل الثالث

بنى التحكم

control structures

3-3-1- بنية الاختيار if :

3-3-2- بنية الاختيار if \else :

3-3-3- بنية الاختيار if \ else المتداخلة

3-3-4- بنية الاختيار switch

3-1- بنى التحكم control structures

يتم عادة تنفيذ تعليمات برنامج ما واحدة تلو الأخرى حسب ترتيب ورودها في نص البرنامج من الأعلى باتجاه الأسفل، نسمي أسلوب التنفيذ هذا بالتنفيذ التسلسلي Sequential execution.

سوف نقوم في الفقرات التالية بعرض العديد من تعليمات C++ التي تساعد المبرمج على تحديد التعليمة التي ستنفذ بعد التعليمة الحالية والتي يمكن أن تأتي في موضع آخر من نص البرنامج مختلف عن موضع التعليمة التالية لها مباشرة، نسمي هذه العملية بعملية نقل التحكم transfer of control ، عملية نقل التحكم هذه ضرورية في حل المشاكل التي تتطلب إجراء اختبارات منطقية لتنفيذ تعليمة معينة وتجاهل أخرى أو العكس، كذلك في تلك التي تتطلب تكرار تعليمة عدداً من المرات طالما أن الشرط محقق أو حتى يتحقق شرط ما (الحلقات).

وقبل الخوض في أنواع بنى التحكم لا بد من التذكير بالمؤثرات العلائقية والمنطقية.

3-3- أنواع بنى التحكم : Types of control structure

من الممكن كتابة البرنامج في لغة C++ باستخدام ثلاث بنى للتحكم هي :

1- البنية التسلسلية Sequence structure:

حيث يقوم الحاسب بتنفيذ تعليمات C++ واحدة بعد الأخرى حسب ترتيب ورودها ضمن نص البرنامج.

2- بنية الاختيار :Selection structure

توفر لغة C++ ثلاثة أنواع من بنى الاختيار:

أ- البنية if : وتسمى بالبنية وحيدة الاختيار.

ب- البنية if/else : وتسمى بالبنية مضاعفة الاختيار كما يمكن أن نجد البنية if/else المتداخلة.

ج- البنية switch وتسمى بالبنية متعددة الاختيار.

3-3-1- بنية الاختيار if :

The if selection structure

تسمى بالبنية وحيدة الاختيار Single selection structure لأنها تسمح باختيار أو تجاهل فعل وحيد.

تملك الصيغة العامة التالية :

```
if ( expression ) statement;
```

حيث أن:

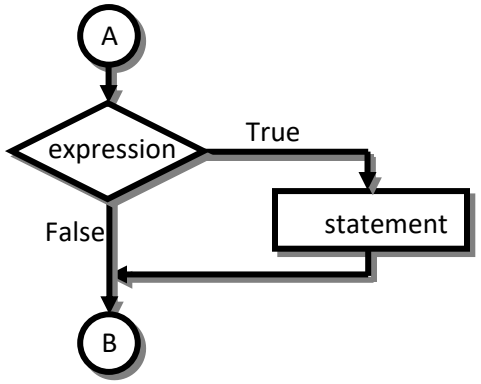
expression: عبارة عن تعبير منطقي أو شرط نتيجة اختباره true أو false.

statement :عبارة عن تعليمة قد تكون بسيطة أو مركبة.

آلية تنفيذ التعليمة :

يتم اختبار expression فإذا كانت نتيجة الاختبار true أو لا تساوي الصفر سيتم تنفيذ statement، أما إذا كانت نتيجة الاختبار false فسيتم تجاهل statement

و يبين الشكل(3-1) المخطط النهجي للبنية if.



الشكل (3-1) المخطط الصندوقي لبنية الاختيار if

مثال 3 -1 :

```
#include<iostream.h>
void main(){
      int x;
      cout<<"enter number:";
      cin>>x;
      if(x>5)cout<<"**"<<endl;
```

```
cout<<"++"<<endl;
cout<<"--"<<endl;}
```

في هذا المقطع البرمجي أعلن عن x بأنه متغير من النوع الصحيح، فإذا تم تنفيذ البرنامج وإدخال قيمة لـ x سنميز حالتين :

أ- إذا كانت قيمة x أكبر من 5 مثلاً 6 :

هذا معناه أن شرط if محقق وبالتالي ستنفذ التعليمة التي بعد if والتي هي في هذا المثال تعليمة بسيطة وسيتم طباعة **، ثم ينتقل الحاسب لتنفيذ التعليمات التالية بشكل تسلسلي حيث يطبع ++ ثم --.

خرج البرنامج :

```
enter number : 6
* *
```

ب- إذا كانت قيمة x أصغر من 5 مثلا 1- :

هذا معناه أن شرط if غير محقق، وبالتالي سيتم تجاهل التعليمة التي بعد if ومتابعة تنفيذ التعليمات التالية بشكل تسلسلي.

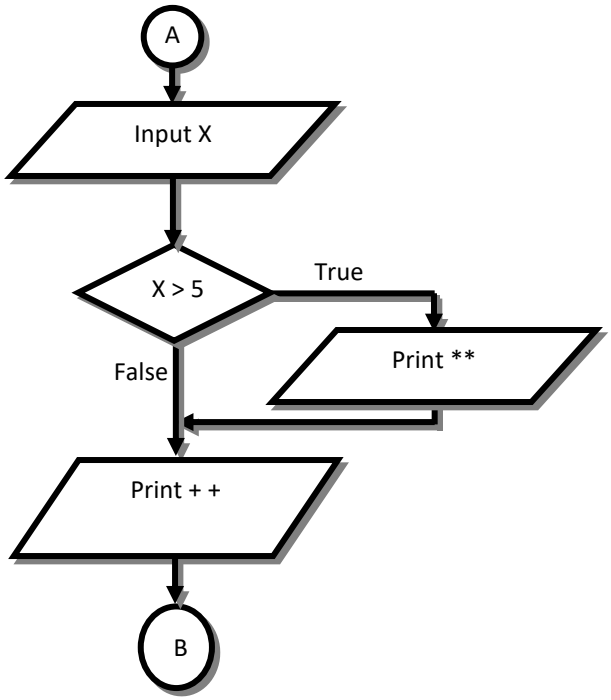
خرج البرنامج :

```
enter number : -1
+ +
```

ويبين الشكل (3-2) المخطط الانسيابي للمثال السابق.

والذي يمكننا التعبير عنه أيضاً باستخدام اللغة الطبيعية كالآتي:

1. البداية.
2. أدخل x.
3. إذا كانت x أكبر من 5:
   - اطبع ** ثم انتقل إلى سطر جديد.
4. اطبع + + ثم انتقل إلى سطر جديد واطبع - -.
5. النهاية.



المخطط الانسيابي(3-2) إذا كانت قيمة x أصغر من 5

مثال 3 -2 :

```
#include<iostream.h>
void main(){
      int x;
      cout<<"enter number : ";
```

```
cin>>x;
if(x>5){
      cout<<"**"<<endl;
      cout<<"++"<<endl;}
cout<<"--"<<endl;}
```

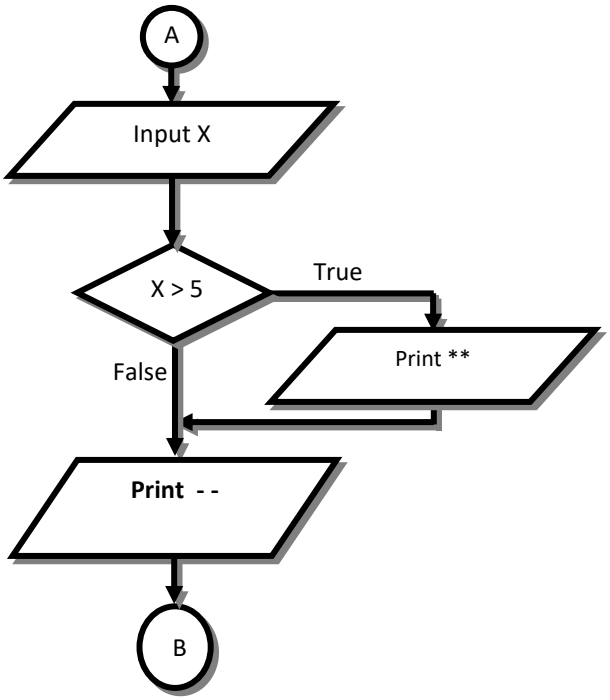
إذا تم إدخال x=7 ثم تم إدخال x=3 سيكون خرج البرنامج :

```
enter number : 7

**

++

--
```

إذا تم إدخال x=3 سيكون خرج البرنامج :

```
enter number : 3

--
```

حيث سيتم تجاهل التعليمة المركبة بعد if والتي تضم تعليمتي طباعة ** و++ لعدم تحقق شرط if.



المخطط الانسيابي(3-3) للمثال 3-2 السابق

مثال 3-3 :

هذا البرنامج يقوم بطباعة أكبر عدد بين ثلاثة أعداد يتم إدخالها إلى البرنامج:

```
#include<iostream.h>
void main(){

      int n1،n2،n3;

      cout<<"enter three integer:";
      cin>>n1>>n2>>n3;
      int max=n1;
      if(n2>max)
            max=n2;
      if(n3>max)
           max=n3;
      cout<<"the maximum is:"<<max;
      cout<<endl;
}
```

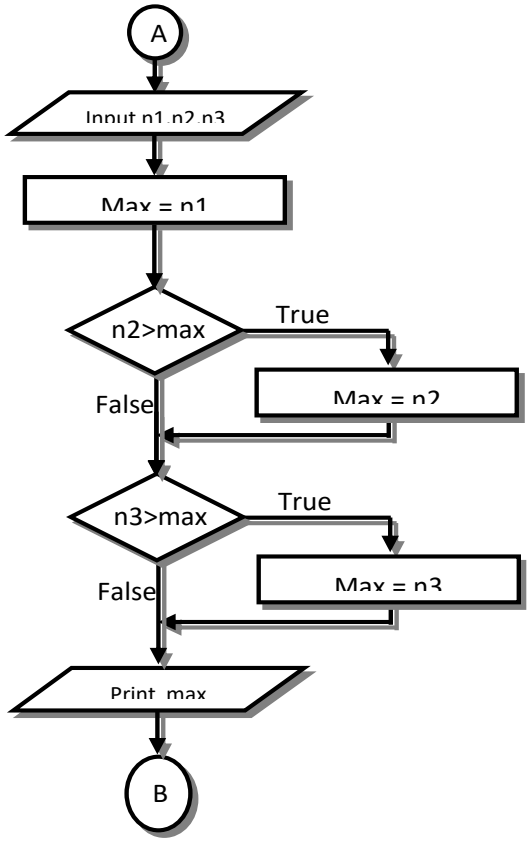
خرج البرنامج :

```
enter three integer : 8  7  9

the maximum is : 9
```

خوارزمية المثال السابق:

تم إدخال القيم:n1=8،n2=7،n3=9

في البداية تم تخصيص قيمة n1 للمتغيرmax أي(max =8) ثم تم اختبار كون n2>max والنتيجة false وبالتالي تم تجاهل التعليمةmax=n2، بعد ذلك تم اختبار كون n3>max وبما أن النتيجة كانت true تم تخصيص قيمة n3 للمتغير max (max=9) ومن ثم طبعت القيمة العظمى، كما هو مبين في الشكل (3-4).



المخطط الانسيابي(3-4) للمثال 3-2

3-2- بنية الاختيار if \else :

The if else selection structure

تسمى هذه البنية بالبنية مضاعفة الاختيار

Double - selection - structure

لأنها تسمح بالاختيار بين فعلين مختلفين.

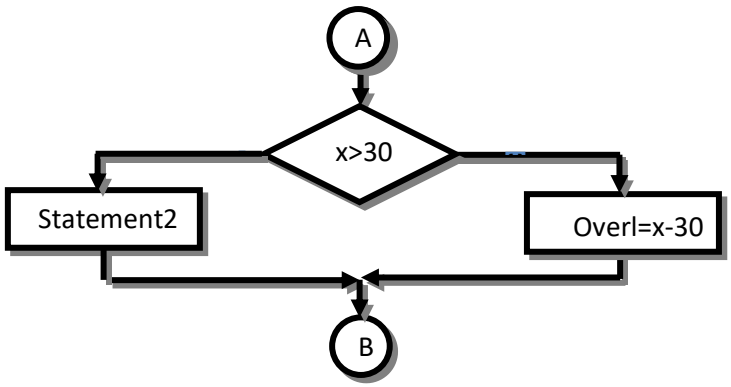
تملك الصيغة العامة التالية :

```
if ( expression ) statement1 ; else  statement 2;
```

آلية تنفيذ التعليمة :

يتم اختبار expression فإذا كانت نتيجة الاختبار true أو لا تساوي الصفر يتم تنفيذ statement1 وتجاهل statement2، أما إذا كانت نتيجة الاختبار false أو تساوي الصفر يتم تجاهل statement1 وتنفيذ statement2.

ويبين الشكل(3-5) المخطط الصندوقي للبنية if/else.



الشكل (3-5) المخطط الصندوقي لبنية الاختيار if /else

هذا ويمكن أن نكتب التعليمة if\else بالشكل التالي :

```
Expression ? statement1:statement2;
```

آلية التنفيذ : إذا كان الشرط محققاً تنفذ statement1 وإلا تنفذ statement2.

مثال :

إذا كان 48 =< grade اطبع " ناجح " وإلا إطبع " راسب ".

```
grade >=48 ?cout <<"passed":cout<<"failed";
```

مثال 3-4

```
#include<iostream.h>
void main(){
      int x;
      cout<<"enter number : ";
      cin>>x;
      if(x>5)
            cout<<"**"<<endl;
      else
            cout<<"++"<<endl;
      cout<<"--"<<endl;
      cout<<"# #"<<endl;}
```

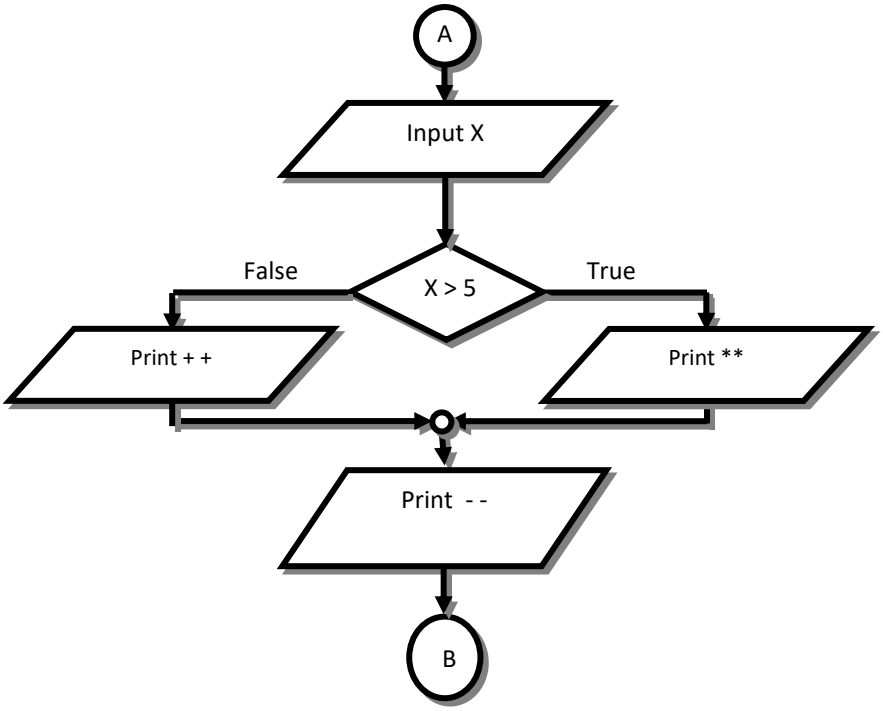
إذا تم تنفيذ البرنامج وأدخلت قيمة لـ x سنميز حالتين:

أ- إذا كان x>5 مثلاً x=7 هذا يعني أن شرط if محقق وبالتالي سيكون خرج البرنامج:

```
enter number : 7
**
--
# #
```

ب- إذا كان `x<5` مثلاً `x = -1` فهذا يعني أن شرط `if` غير محقق وبالتالي سيكون خرج البرنامج :

```
enter number : -1

+ +

- -

# #
```



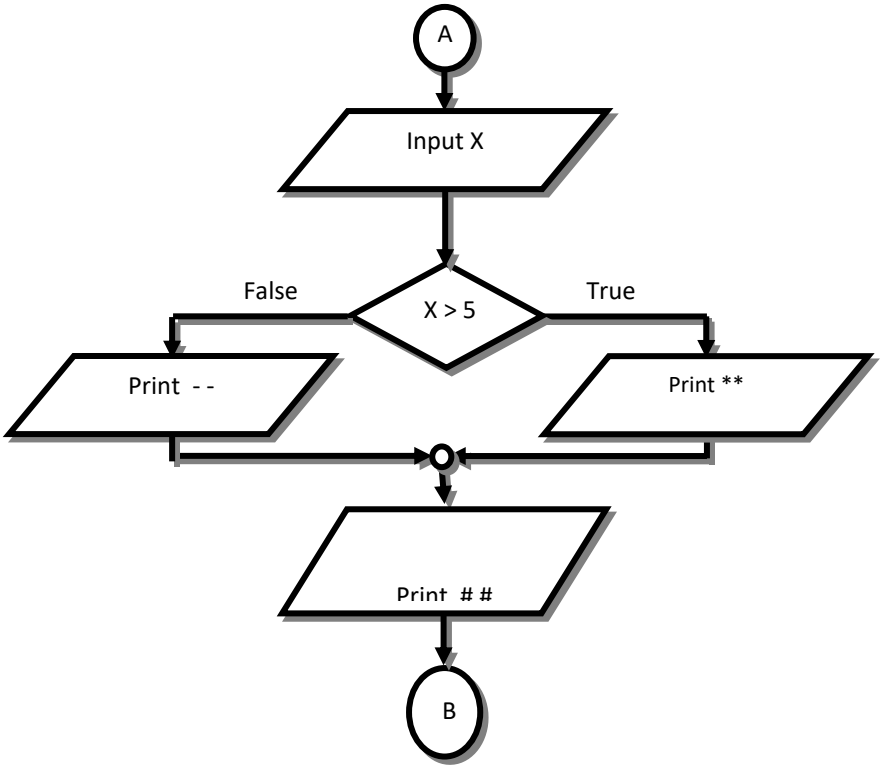
المخطط الانسيابي(3-6) للمثال 3-4

مثال 3-5 :

```
#include<iostream.h>
void main(){
      int x;
      cout<<"enter number : ";
      cin>>x;
      if(x>5){
                  cout<<"**"<<endl;
                  cout<<"++"<<endl;}
      else
            cout<<"--"<<endl;


      cout<<"##"<<endl;}
```

إذا كان x>5 فسيكون خرج البرنامج :

```
enter number : 7

* *

+ +

# #
```



المخطط الانسيابي(3-7) للمثال 3-5

وذلك لتنفيذ التعليمة المركبة بعد `if` وإهمال تلك التي بعد `else`.

إذا كان `x<5` سيكون خرج البرنامج :

```
enter number : -1

- -

# #
```

3-3-3- بنية الاختيار `if \ else` المتداخلة

`Nested if else selection structure`

يتم اختبار عدة تعابير منطقية أو شروط وينتهي العمل عند إحداها.

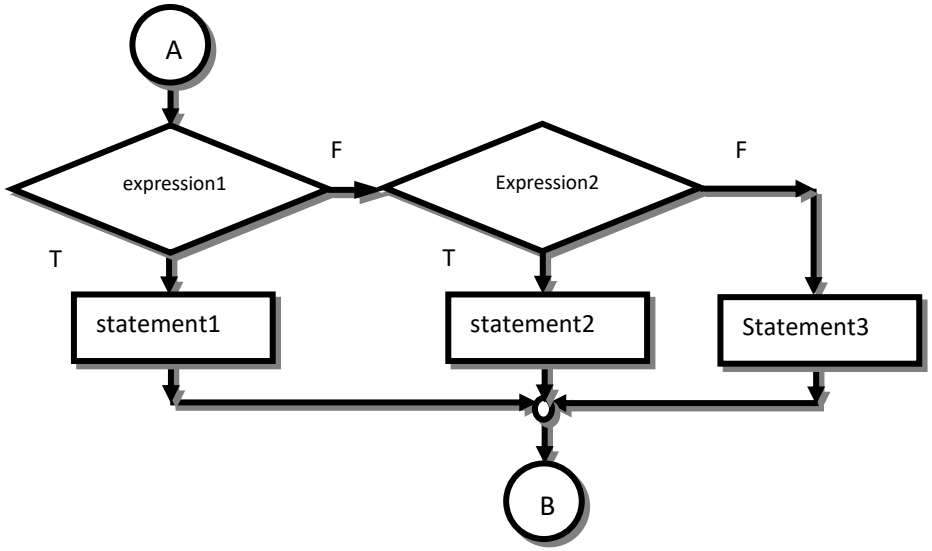
تملك الصيغة العامة التالية :

```
if (expression 1) statement 1; else
if (expression 2) statement 2 ; else
if (expression 3) statement 3……..
```

آلية تنفيذ التعليمة :

يتم اختبار expression1 فإذا كانت نتيجة الاختبار true أو لا تساوي الصفر يتم تنفيذ statement1 وتجاهل باقي البنية if \ else المتداخلة ولكن إذا كانت نتيجة الاختبار false أو تساوي الصفر يتم تجاهل statement1 وتنفيذ ما بعد else حيث يتم اختبار expression2 وهكذا دواليك.

ويبين الشكل (3-8) المخطط الصندوقي للبنية if\else المتداخلة.



الشكل (3-8) المخطط الصندوقي لبنية الاختيار if /else المتداخلة

مثال 3-6 :

في هذا المثال سيتم إدخال علامة الطالب من خلال المتحول الصحيح grade فإذا كانت العلامة المدخلة أكبر أو مساوية لـ90 سيتم طباعة المحرف A وتجاهل باقي تعليمات البرنامج.

في حين أنه يطبع B إذا كانت العلامة بين 80 و90 .

وسيطبع C إذا كانت العلامة المدخلة بين 70 و79.

وسيطبع M إذا كانت العلامة المدخلة بين 60 و69.

و سيطبع f إذا كانت العلامة المدخلة أصغر من 60.

```
#include<iostream.h>
void main(){
      int grade;  cout<<"enter grade:";
      cin>>grade;
      if(grade >=90)
          cout<<"A"<<endl;
      else
          if(grade >=80)
                    cout<<"B"<<endl;
          else
               if(grade >= 70)
                    cout<<"C"<<endl;
               else
               if(grade >= 60)
                        out<<"M"<<endl;
                  else
                cout<<"F"<<endl;}
```

خرج البرنامج

```
enter grade: 75

C

Press any key to continue
```

تسمى بالبنية متعددة الاختيار multiple - selection - structure لأنها تسمح باختيار فعل محدد من بين مجموعة ممكنة من الأفعال المختلفة.

تملك الصيغة العامة التالية :

```
switch (expression)
```

```
 {
case constant 1: statement_1; break;
case constant 2: statement_2; break;
case constant 3: statement_3; break;
.
.
default : statement_n ;}
```

آلية تنفيذ التعليمة :

تبدأ بنية الاختيار switch بالكلمة المفتاحية switch.

Expression: هو عبارة عن تعبير يأخذ قيماً صحيحة أو محرفية كذلك الأمر بالنسبة للثوابت constant.

الأمر switch يحدد قيمة التعبير expression فإذا كانت تلك القيمة مطابقة لقيمة أحد الثوابت في الحالات case يتم تنفيذ التعليمة statement (مركبة أو بسيطة) والمرتبطة بتلك الحالة ثم يتم الخروج من البنية switch.

إذا لم يحدث أي تطابق بين قيمة expression وقيمة أحد الثوابت في حالات case يتم تنفيذ التعليمة المرتبطة بـ default ومن ثم الخروج من switch، ولكن وفي هذه الحالة وإذا لم تكن default موجودة سيتم الخروج من switch دون تنفيذ أي من تعليماتها.

إن عملية الخروج من switch تؤمن بواسطة التعليمة break التي تسبب نقل التحكم خارج البنية وسندرسها فيما بعد.

مثال 3-7 : سيتم إدخال أحد رموز العمليات الأساسية وإجراء هذه العملية على العددين x=10، y=5 حيث تستخدم عبارة switch في هذا المثال ونكتب فيما يلي الخوارزمية المطبقة معبراً عنها باللغة الطبيعية:

1. البداية
2. إسناد قيم x،y .

3. إدخال رمز العملية ch.

4. الدخول في عبارة switch وفق هذا الرمز.

- إذا كانت ch هي رمز الجمع عندها يتم طباعة قيمة =x+y ثم ناتج الجمع ثم الانتقال إلى 4.
- وإلا إذا كانت ch هي رمز الطرح عندها يتم طباعة قيمة =x-y ثم ناتج الطرح ثم الانتقال إلى 4.
- وإلا إذا كانت ch هي رمز الضرب عندها يتم طباعة قيمة =x*y ثم ناتج الضرب ثم الانتقال إلى 4.
- وإلا إذا كانت ch هي رمز القسمة عندها يتم طباعة قيمة =x/y ثم ناتج القسمة ثم الانتقال إلى 4.
- وإلا إذا كانت ch هي رمز باقي القسمة عندها يتم طباعة قيمة =x%y ثم ناتج باقي القسمة ثم الانتقال إلى 4.

5. النهاية.

```
#include<iostream.h>


void main()
{
   char ch; int x=10;int y=5;
   cout<<"enter mathematical operations:";


   cin>>ch;
   switch (ch)
         {
         case'+':
               cout<<x<<'+'<<y<<'='<<x+y<<endl;
               break;
```

```
case '-':
      cout<<x<<'-'<<y<<'='<<x-y<<endl;
      break;
case '*':
      cout<<x<<'*'<<y<<'='<<x*y<<endl;
      break;
case '/':
      cout<<x<<'/'<<y<<'='<<x/y<<endl;
      break;
case '%':
      cout<<x<<'%'<<y<<'='<<x%y<<endl;
      break;      }    }
```

إذا تم إدخال المحرف + سيتم إيجاد مجموع العددين y،x، أما إذا تم إدخال المحرف – سيتم إيجاد فرق العددين y،x وهكذا، بالتالي يقوم البرنامج مقام آلة حاسبة بسيطة.

خرج البرنامج

```
enter  mathematical operations :*

10 * 5 = 50
```

وإذا تم إدخال محرف مخالف للمحارف المذكورة فلن نحصل على خرج.

```
enter  mathematical operations :=

Press any key to continue
```

ملاحظة 1 :

إن ثوابت الحالة case محارف لذلك وجب وضعها ضمن فاصلتين علويتين.

ملاحظة 2 :

يمكن إعطاء رسالة خطأ عند إدخال محرف غير مناسب بإدخال الجزء default في البرنامج بالشكل :

```
default  : cout << " error !! enter a new char " << endl
;
```

ملاحظة 3 :

إذا كان للثوابت المختلفة لحالات case نفس التعليمات يمكن أن يستبدل الشكل :

```
case  constant 1 : statement1 ; break;
case  constant 2 : statement1 ; break;
case  constant 3 : statement2 ; break;
case  constant 4 : statement2 ; break;
   .
```

بالشكل :

```
case  constant 1 : constant 2 : statement 1 ; break;
case  constant 3 : constant 4 : statement 2 ; break;
   .
   .
```

مثال 3-8 :

هذا البرنامج يطبع red إذا كان المحرف المدخل r أو R، في حين يطبع white إذا كان المحرف المدخل w أو W، ويطبع blue إذا كان المحرف المدخل b أو B ويطبع رسالة error char إذا كان المحرف المدخل غير ذلك.

```
#include<iostream.h>
void main()
{
      char choice;
      cout<<"enter a char:";
      cin>>choice;
      switch (choice)
            {
            case 'r':case 'R':
                  cout<<"red"<<endl;break;
            case 'w':case 'W':
                  cout<<"white"<<endl;break;
```

```
            case 'b':case 'B':
                  cout<<"blue"<<endl;break;
            default:
                  cout<<"error char"<<endl;break;
      }}
```

خرج البرنامج

```
enter a char : r

red
```

إذا كانت الثوابت constant أعدادا صحيحة (ليست محارف) عندئذ لا توضع هذه الثوابت ضمن فواصل علوية.

مثال 3-9 :

يمكن كتابة البرنامج في المثال 3-6 باستخدام البنية switch كما يلي :

```
#include<iostream.h>
void main()
{
      int score;
      cin>>score;
      switch (score/10)
      {
            case 10:case 9:
                  cout<<'A'<<endl;break;
            case 8:cout<<'B'<<endl;break;
            case 7:cout<<'c'<<endl;break;
            case 6:cout<<'M'<<endl;break;
            case 5:case 4:case 3:case 2:
            case 1:cout<<'F'<<endl;break;
            default:cout<<"error score."<<endl;}}
```

يتم إدخال علامة الطالب في المتحول score، ثم تقسم على10، طالما أن معامل switch صحيح سيتم أخذ القسم الصحيح من ناتج القسمة.

مكتبة A to Z