



المعلوماتية كلية الهندسة

مدخل إلى الخوارزميات والبرمجة

Introduction to Algorithms and Programming

ا. د. علي عمران سليمان

محاضرات الأسبوع السادس

الفصل الثاني 2022-2023

بنى التحكم control structures

3-3-2- بنى الاختيار if/else	3-3-1- بنى التحكم control structures
3-3-3- بنى الاختيار if / else المتداخلة	3-2- المؤثرات العلائقية والمنطقية .
3-3-4- بنى الاختيار switch	Logical and relational operators
3-3-5- البنية التكرارية while (حلقة while)	3-3- أنواع بنى التحكم .
3-3-6- البنية التكرارية do \ while	أ- البنية التسلسلية Sequential structure
3-3-7- البنية التكرارية for	ب- بنى الاختيار Selection structure
3-3-8- الحلقات المتداخلة	ت- بنى الشرط:
3-4- التعليمتان continue, break	3-3-1- بنى الاختيار if :

المحاضرة من المراجع :

[1]- Deitel & Deitel, C++ How to Program, Pearson; 10th Edition (February 29, 2016)

[2] - د. علي سليمان, مدخل إلى الحاسوب والخوارزميات, جامعة تشرين 2005-2006

3-1- بني التحكم control structures

- **التنفيذ التسلسلي Sequential execution:** هو تنفيذ تعليمات برنامج ما واحده تلي الأخرى حسب ترتيب ورودها في نص البرنامج من الأعلى باتجاه الأسفل، وهذا ما يعرف بالتحكم الافتراضي.
- **التحكم:** هو 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**.
- **يستخدم نقل التحكم:**
- عند الاختبارات العلائقية او المنطقية التي تتطلب تنفيذ تعليمة معينة وتجاهل أخرى أو العكس.
- الحلقات التي تتطلب تكرار تعليمة عدداً من المرات طالما أن الشرط محقق.
- **تنفيذ التعليمتان break, continue.**

Logical and relational operators

3-2- المؤثرات العلائقية والمنطقية Logical and relational operators

أ- تسمح المؤثرات العلائقية **Relational operators** بمقارنة قيمتين أو عبارتين تؤديان إلى قيمة بوليانية **true** أو **false**، حيث الصفر هو **false** وغير ذلك هو **true**، وهذه المؤثرات هي:

>, >=, <, <=, ==, !=

`count <=100;` `sqrt (a+b+c) > 0.005;` `letter != 'x';`

ب - المؤثرات المنطقية **Logical operators**: تتعامل المؤثرات المنطقية مع القيم المنطقية **true** أو **false** وهي: **&&** (and), **||** (or) ومؤثر النفي الأحادي **!** (not).

إن **and, or** تحتاج عاملين وتستخدمان لدمج تعابير منطقية، أما **not** مؤثر احادي يعمل لقلب حالة العامل.

`(count <=100) &&(ch1 != '*');` `(answer <0) || ((answer >5.0) && (answer <=10.0));`

بنى التحكم ثلاث :

- البنى التسلسلية Sequential structure

- البنى الاختيارية Selection structure

(if, if else, if else if else if else if else , switch)

- البنى التكرارية Loop structure

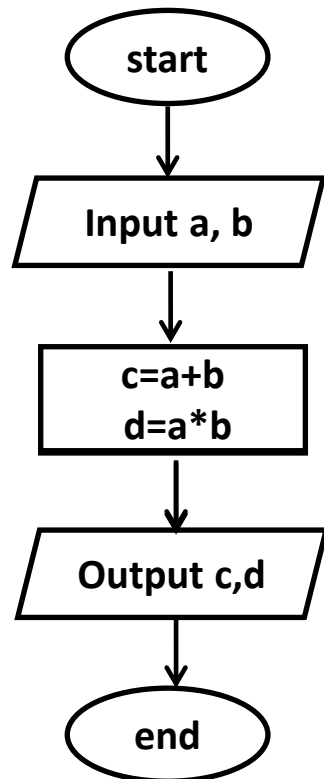
(while, for, do while)
break, continue

- تعليمتي الانتقال

3-3-1- البنى التسلسلية Sequential structure (1)

المخطط التدفقي التتابعي البسيط: مسائل لا يقتضي حلها أية تجاهل أو تكرار للخطوات بل كل خطوه تنفذ مره واحده وواحد فقط.

عودة للمثال 1: المخطط التدفقي التتابعي البسيط. ليكن المطلوب رسم المخطط التتابعي وكتابة الخوارزمية والكود لإدخال عددين وحساب مجموعهما وجدائهما وطباعة الناتج.



```

#include "stdafx.h"
#include<iostream>
using namespace std;
void main()
{
    int a, b, c, d;
    cin>>a>>b;
    c=a+b; d=a*b ;
    cout<<"c= "<<c<<endl;
    cout<<"d= "<<d<<endl;
}
  
```

Text Algorithm (Pseudo code)

1. Start
2. Input a, b
3. Set c=a +b and Set d= a*b
4. Output c, d
5. end

شرح الكود البرمجي أو البرنامج

```
#include "stdafx.h"  
#include<iostream>  
using namespace std;  
void main()  
{  
    int a, b, c, d;  
    cin>>a>>b;  
    c=a+b;  
    d=a*b ;  
    cout<<"c= " <<c<<endl;  
    cout<<"d= " <<d<<endl;  
}
```

وجود # يدل على أنه ملف رأسي يتم تفعيله ما قبل الترجمة.

"#include "stdafx.h" يؤمن عدم تكرار مطابقة ماتم نجاح مطابقته سابقاً

<iostream>#include يؤمن إمكانية استخدام معاملات وتوابع للدخل والخرج

using namespace std; توجيه المترجم لاستخدام مساحات الأسماء القياسية لتنظيم التعليمات البرمجية في مجموعات منطقية ولمنع تضارب الأسماء التي يمكن أن تحدث عندما تتضمن قاعدة البيانات الخاصة بك عدة مكتبات.

السطر الأول رأس التابع الرئيس main لا يعيد شيئاً لوجود void قبل الاسم ولا يرسل له شيئاً () لفراغ القوسين.

{ القوسان وما بينهما يعبران عن جسم التابع الرئيس ويتكون من:

- تعريف كل المتغيرات التي ستستخدم في الدخل والخرج والحساب

- إدخال المعطيات a, b

- حساب المطلوب c, d

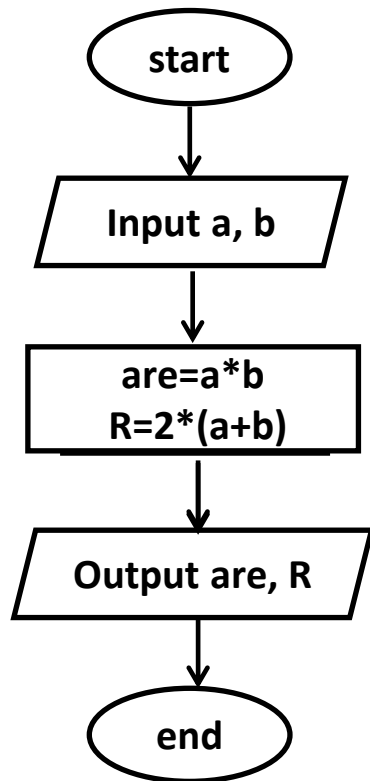
- طباعة النتائج المحسوبة كل على سطر مثلاً مع تعريف بسيط بها.

3-3-1- البنى التسلسلية

Sequential structure (2)

بعض المسائل لا يقتضي حلها أية تجاهل أو تكرار للخطوات بل كل خطوه تنفذ مره واحده وواحد فقط

عوده للمثال 2: المخطط التدفقي التتابعي البسيط. ليكن المطلوب كتابة ورسم المخطط التتابعي والكود البرمجي لحساب مساحة ومجيط الشكل الرباعي الذي بعده A & B وإخراج النتائج.



Flow Chart

```

#include "stdafx.h"
#include<iostream>
using namespace std;
void main()
{
    int a, b, r, are;
    cin>>a>>b;
    r=2*(a+b);
    are=a*b ;
    cout<<"r= "<<r<<endl;
    cout<<"area= "<<are<<endl;
}
  
```

Pseudo code

Text Algorithm (Pseudo code)

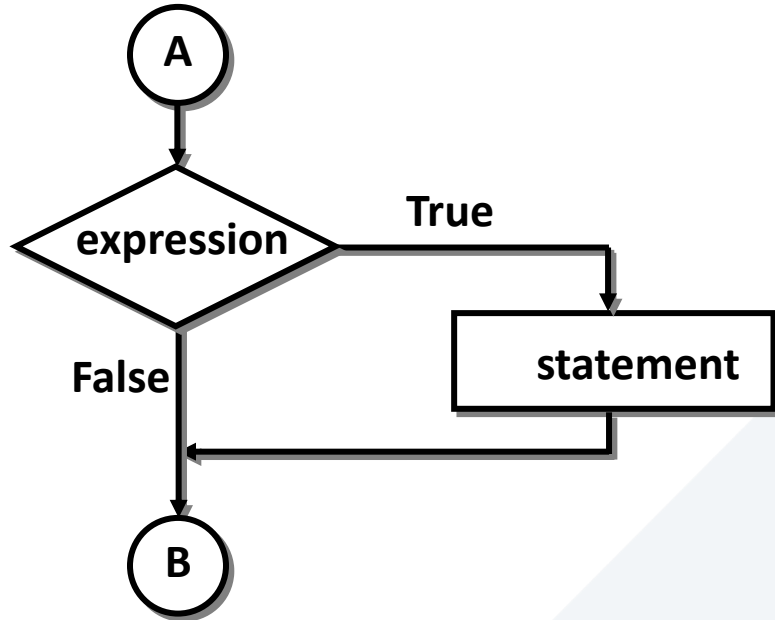
1. Start
2. Input a, b
3. Set $R=2*(a + b)$ and Set $are= a*b$
4. Output are, R
5. end

if, if else, if else if else if else if else , switch, break, continue

- أ- البنية if : وتسمى بالبنية وحيدة الاختيار.
- ب- البنية if/else : وتسمى بالبنية مضاعفة الاختيار.
- ت- البنية if/else if ... else متعددة الاختيار أو المتداخلة (المشعبة).
- ث- البنية switch وتسمى بالبنية متعددة الاختيار.
- ج- التعليمتان break, continue .

البنية وحيدة الاختيار if

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



الصيغة العامة
if (expression) statement;

if: كلمة محجوزة لغايتها.

expression: عبارة عن تعبير منطقي أو شرط يوضع بين قوسين نتيجة اختبار true أو false.

statement: عبارة عن تعليمة قد تكون بسيطة أو مركبة. آلية تنفيذ التعليمة :

يتم اختبار expression فإذا كانت نتيجة الاختبار true (أي لا تساوي الصفر) سيتم تنفيذ statement والمتابعه في باقي البرنامج، أما إذا كانت نتيجة الاختبار false فسيتم تجاهل statement والمتابعه في باقي البرنامج.

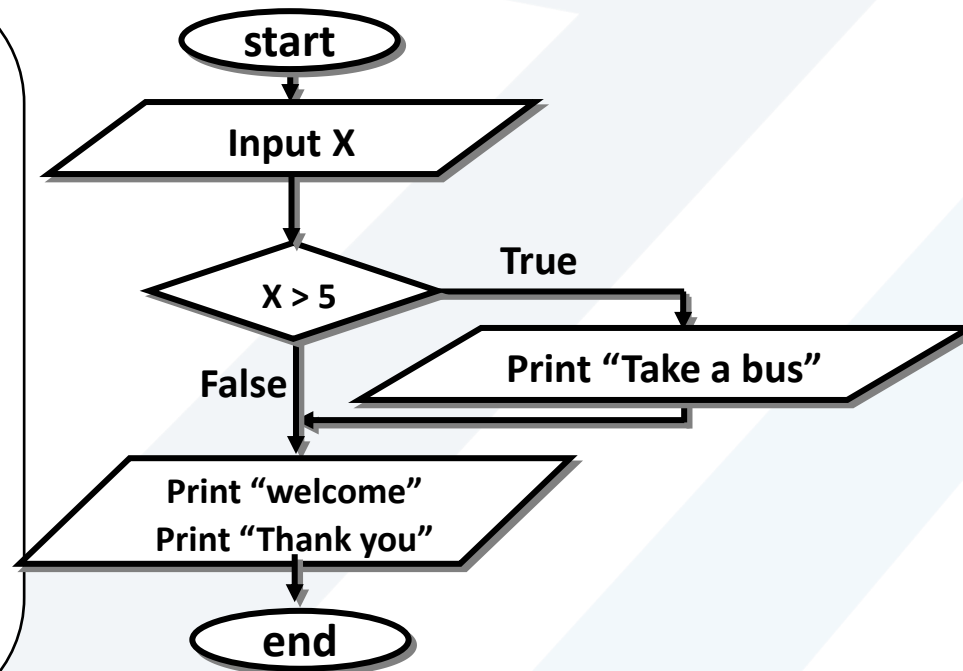
ليكن المطلوب إدخال عدد إذا كان أكبر من 5 يطبع Take a bus ثم welcome ثم Thank you كل على سطر منفرد وإلا لا يطبع Take a bus ويطبع الباقي

program code

flow chart

Text Algorithm

```
#include "stdafx.h"
#include<iostream>
using namespace std;
void main()
{
    int x;
    cout<<"enter number:";
    cin>>x;
    if(x>5)
        cout<<"Take a bus\n" ;
    cout<<" welcome " <<endl;
    cout<<" Thank you " <<endl;
}
```



Text Algorithm (Pseudo code)

1. Start
2. Input x
3. if (x>5) then
 1. Output "Take a bus"
 2. new line
4. Output "welcome " new line
5. Output "Thank you "
6. end

لاحظ عدم وجود **else** وسنرى لاحقاً استخدام **if/else if else** المتداخلة لحل التمرين

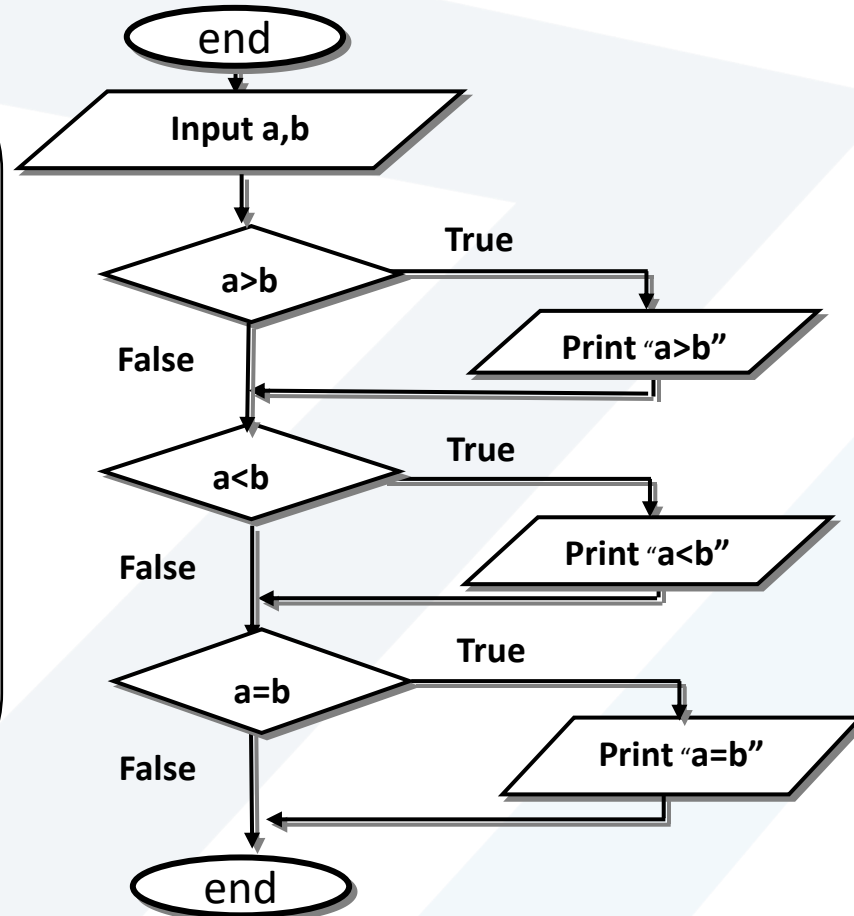
مثال 9: ليكن المطلوب إدخال عددين وإخراج الأكبر أو التساوي بينهما.

program code

flow chart

Text Algorithm

```
#include "stdafx.h"
#include<iostream>
using namespace std;
Void main()
{
    int a, b;
    cout<<" a="; cin>>a;
    cout<<" b="; cin>>b;
    if ( a > b ) cout << "a>b";
    if ( a < b ) cout <<"a<b" ;
    if ( a == b ) cout << "a=b" ;
}
```



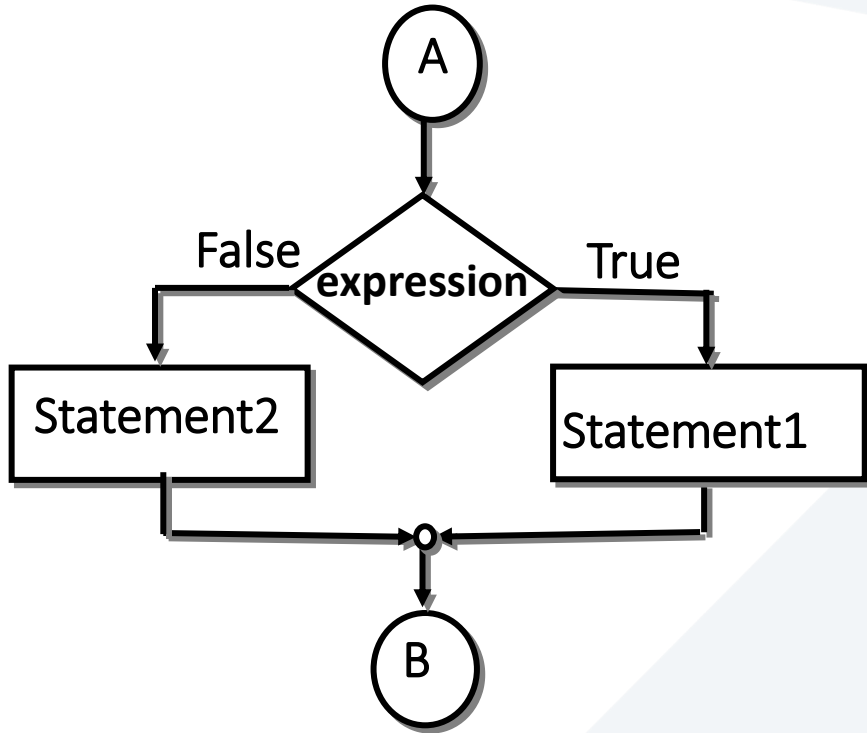
Text Algorithm (Pseudo code)

1. Start
2. Input a, b
3. If (a > b) then
 1. Output "a>b"
4. If (a < b) then
 1. Output "a<b"
5. If (a = b) then
 1. Output "a=b"
6. end

Note: We use if sequences, so we need to test for equality.

3-3-2- بنية الاختيار if/else

البنية بخيارين Double selection structure لأنها تسمح باختيار الأول وتجاهل الثانية أو العكس.



الصيغة العامة

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

أو تكتب `Expression ? statement1:statement2;`

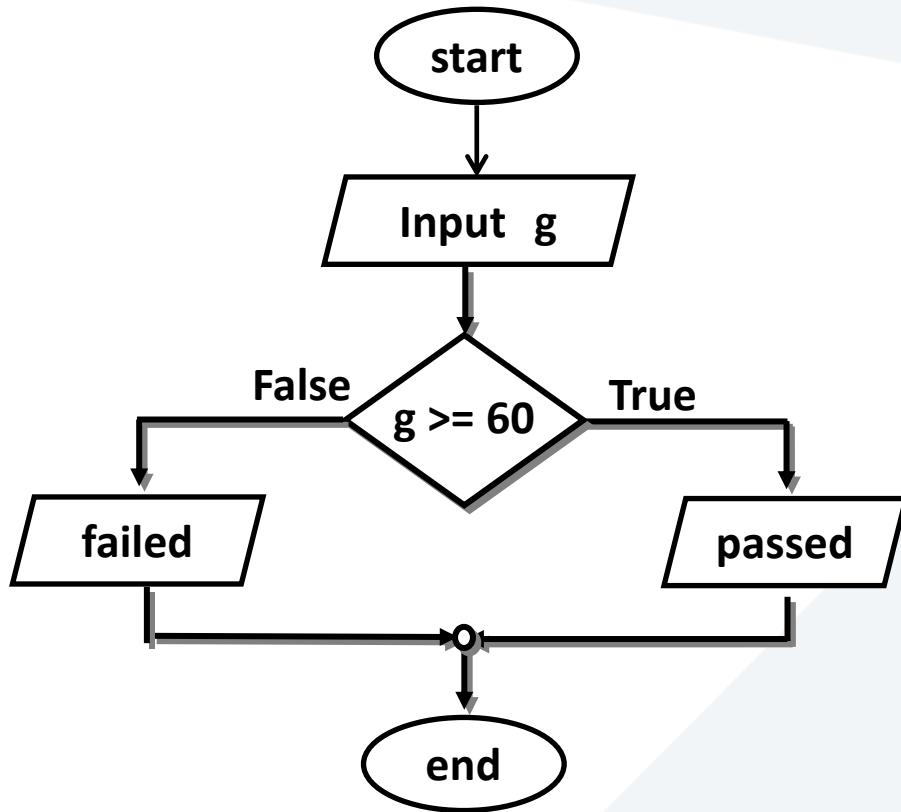
expression: تعبير منطقي أو شرط نتيجة اختباره `true` أو `false`.
statement1, statement2: عبارة عن تعليمه بسيطة تنتهي بـ ; وإذا كانت أكثر من واحدة أي مركبة توضع ما بين {} لتعامل معاملة البسيطة.

آلية تنفيذ التعليمة `if \else`:

يتم اختبار `expression` فإذا كانت نتيجة الاختبار `true` سيتم تنفيذ `statement1`، وتجاهل `statement2` أما إذا كانت نتيجة الاختبار `false` فسيتم العكس تجاهل `statement1`، وتنفيذ `statement2`.

3-3-2- بنية الاختيار if /else

مثال 10: ليكن المطلوب إدخال درجة طالب grade وإذا كانت $60 \leq \text{grade}$ اطبع passed وإلا اطبع failed.



```

#include "stdafx.h"
#include<iostream>
using namespace std;
void main()
{
    int g;
    cin>>g;
    if(g>=60)
        cout<<"passed "<<endl;
    else cout<<"failed "<<endl;
    /* g >=60?cout << "passed" :
    cout<<"failed"; */
}
  
```

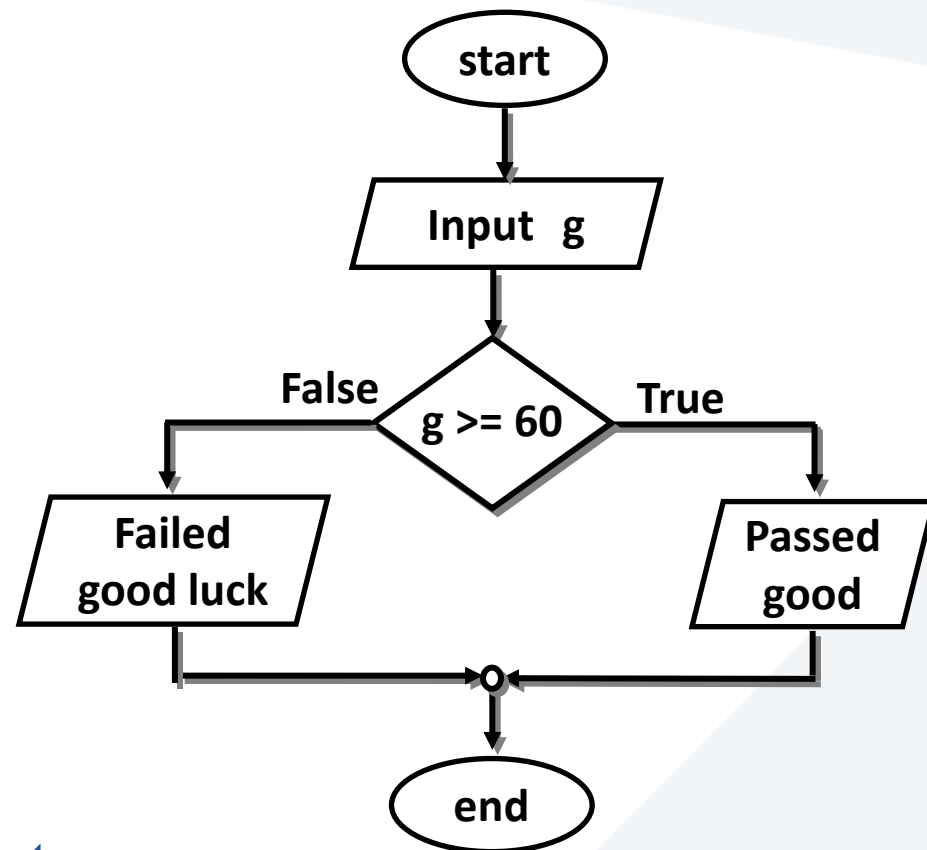
Text Algorithm (Pseudo code)

1. Start
2. Input g
3. If (g>=60)then
 output "passed"
 else
 output "failed"
4. end

يمكن استبدال التعليمة if بالتعليمة الموجودة كتعليق مابين /* */ لتعطي الناتج ذاته

if /else بنية الاختيار 2-3-3

ليكن المطلوب إدخال درجة طالب g وإذا كانت $g \geq 60$ اطبع `passed` و `good` وإلا اطبع `failed` و `good luck` كل كلمة تطبع على سطر منفرد.



```

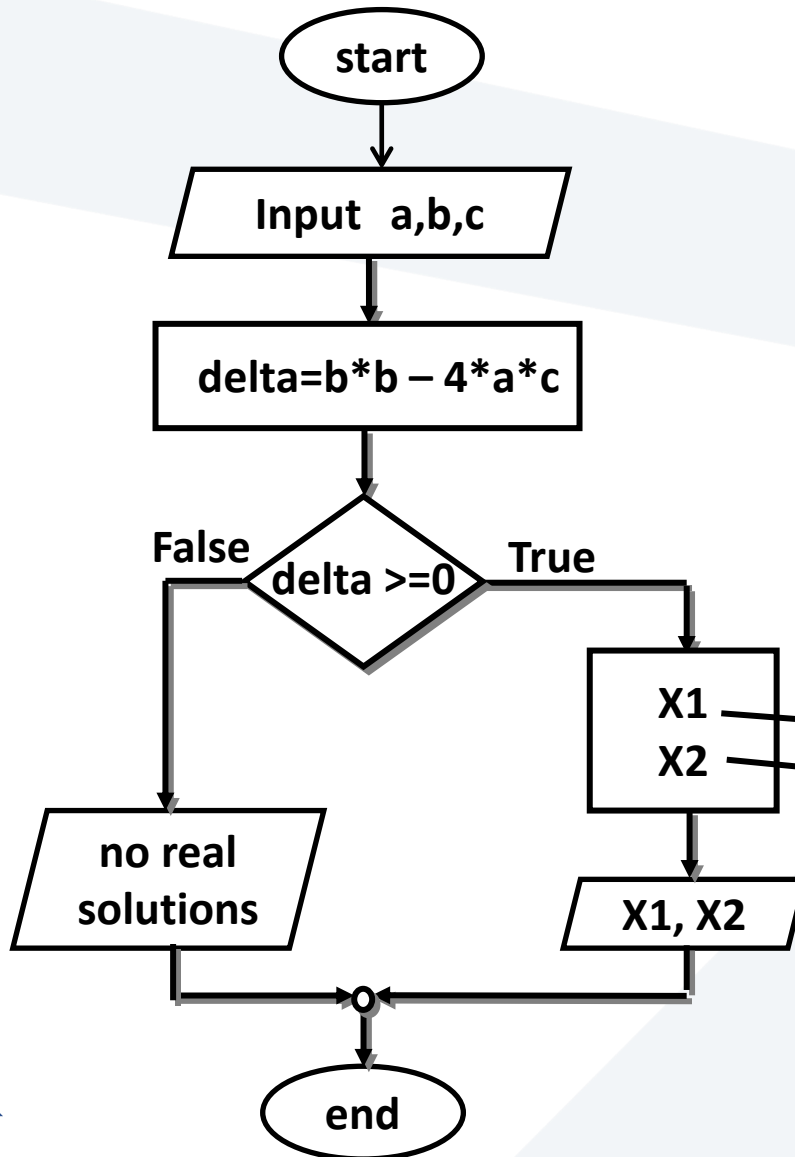
#include "stdafx.h"
#include<iostream>
using namespace std;
void main()
{
  int g; cin>>g;
  if(g>=60)
    cout<<" passed \n
    good" <<"\n";
  else
    {cout<<"failed "<<endl;
    cout<<"good luck "<<endl ;
    }
}
  
```

Text Algorithm (Pseudo code)

1. Start
2. Input g
3. If $(g \geq 60)$ then
 - output "passed"
 - new line output "good"
 - else
 - new line output "failed"
 - new line output "good luck"
4. end .

إن "\n" تعادل endl حالة نعم
في حالة else نظراً لوجود تعليمتين يتطلب
الأمر تحويلهما لتعليمة مركبة باستخدام { }

مثال 11: ليكن المطلوب إيجاد الجذور الحقيقية لمعادلة من الدرجة الثانية وطباعتها إن أمكن وإلا طباعة no real solutions.



```

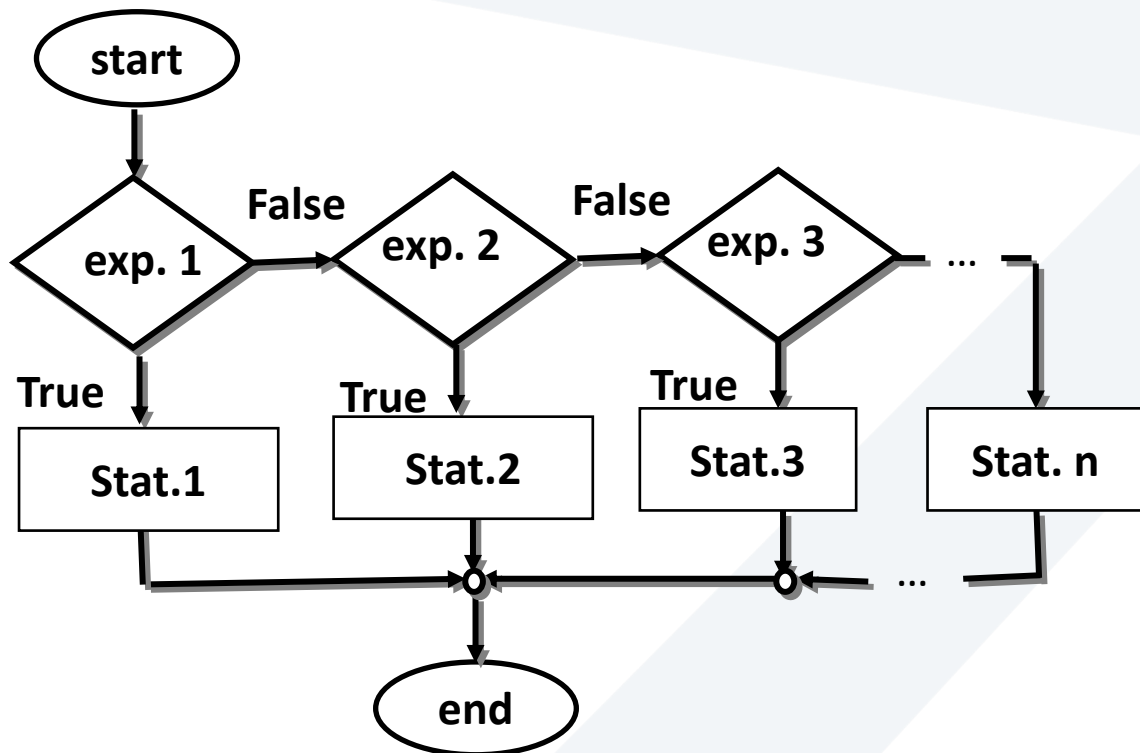
#include "stdafx.h"
#include<iostream>
#include<math.h>
using namespace std;
void main()
{
    int a,b,c ,delta, x1, x2;
    cin>>a>>b>>c;
    delta=b*b - 4*a*c;
    if(delta >=0)
    {
        x1=(b+sqrt(delta))/(2*a);
        x2=(b-sqrt(delta))/(2*a);
        cout<<"x1= "<<x1<<
        "\nx2= "<<x2<<"\n";
    }
    else
        cout<<"no real solutions\n";
}
  
```

Text Algorithm (Pseudo code)

1. Start
2. Input a, b, c
3. Set $\text{delta} = b^2 + 4*a*c$
4. If ($\text{delta} \geq 0$) then
 1. Set $X_1 = (b + (\text{delta})^{0.5}) / (2*a)$
 2. Set $X_2 = (b - (\text{delta})^{0.5}) / (2*a)$
 3. Output X_2, X_1
5. end

نظراً لقلّة الفراغات تم الدلالة لحساب x_1, x_2 من الكود البرمجي فقط

3-3-3- بنية الاختيار if / else المتداخلة - Nested if else selection structure



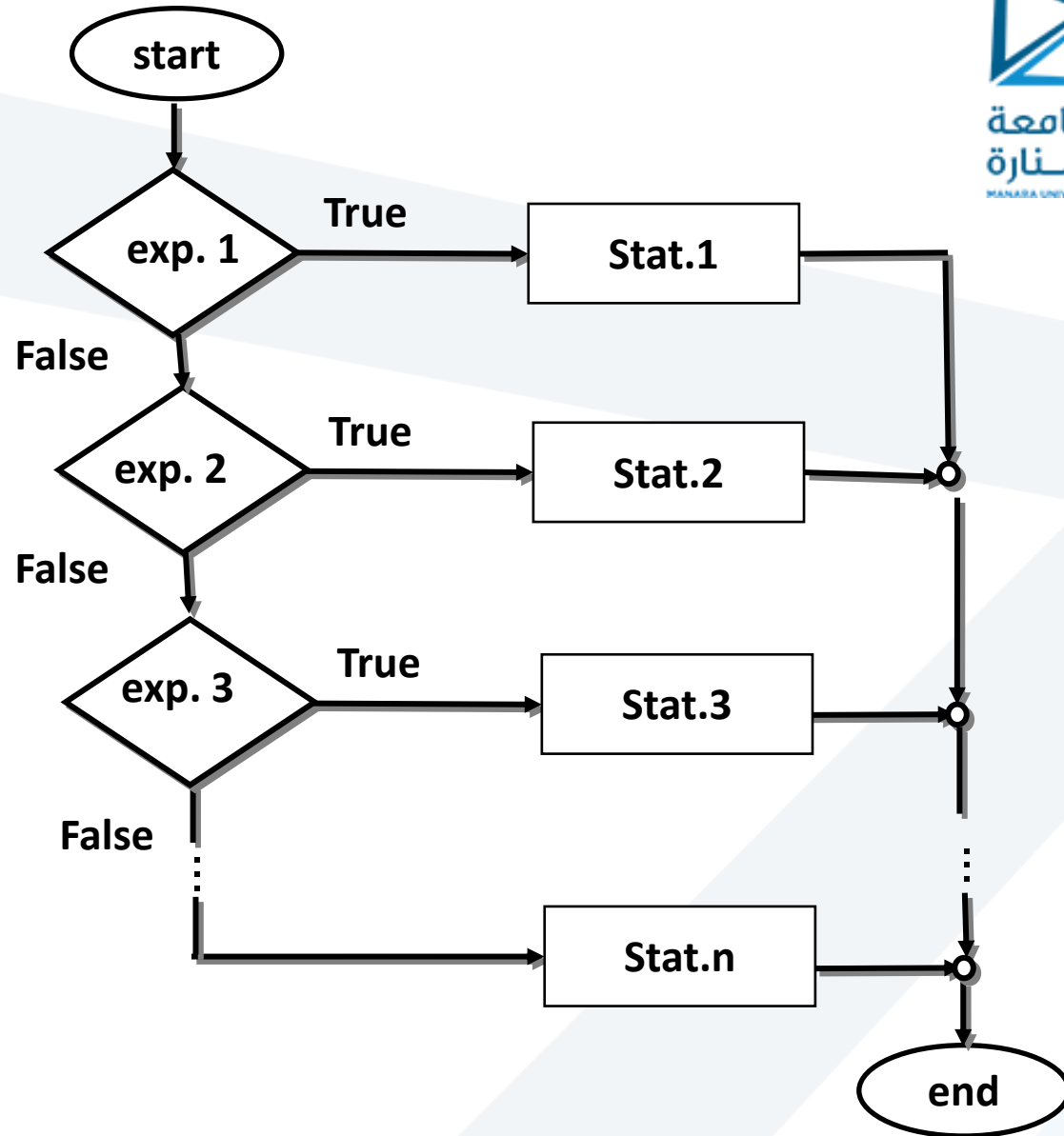
الصيغة العامة (متعدد الاختيار)

```

if (expression1) statement1;
else if (expression2) statement2 ;
else if (expression3) statement3; ...
else statement4;
  
```

expression1: عبارة عن تعبير منطقي أو شرط إذا كان **true** ننفذ **statement1** كانت عبارة بسيطة أو مركبة ونتجاهل الباقي، وإن كانت **false** نتجاهل **statement1** ونختبر **expression2** إذا كان **true** ننفذ **statement2** ونتجاهل الباقي، وإن كانت **false** نتجاهل **statement2** ونختبر **expression3** وهكذا ... في حال كل العبارات الشرطية **False** سيتم تنفيذ **statement n**.

بنية الاختيار if / else المتداخلة-2-

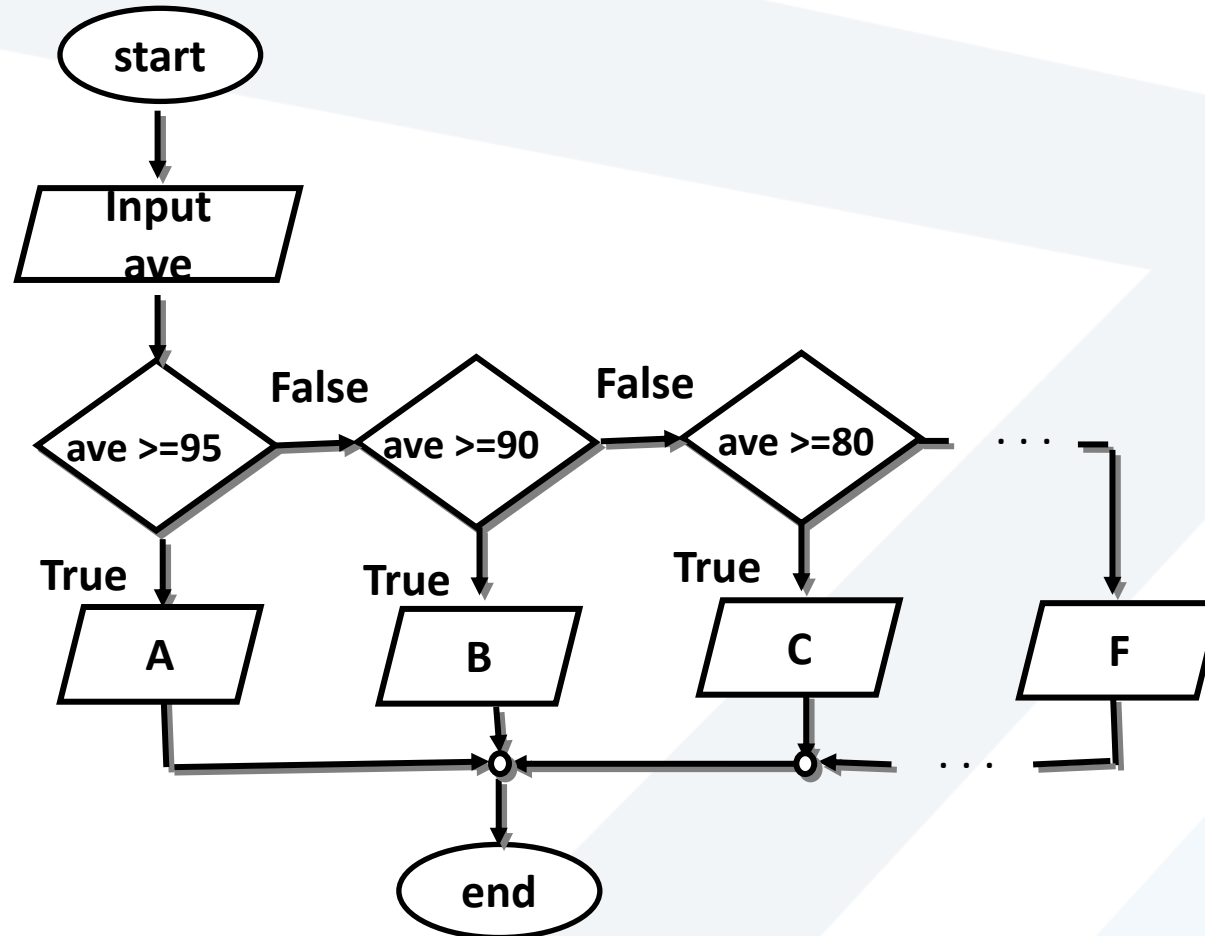


وقد ترسم بشكل مكافئ
وتعبر Stat.n عن الحالة الافتراضية

بنية الاختيار المعشعبة



مثال 12: بنية الاختيار المعشعبة ... if else if else if ...
else ليكن المطلوب إدخال معدل الطالب من خلال المتحول
الصحيح ave ومعرفة تصنيف الطالب كما درسنا سابقاً.



Text Algorithm (Pseudo code)

1. Start
2. Input x
3. if $(x \geq 95)$ then
 Output 'A'
else
 if $(x \geq 90)$ then
 Output 'B'
 else
 if $(x \geq 80)$ then
 Output 'C'
 else
 if $(x \geq 70)$ then
 Output 'D'
 else
 if $(x \geq 60)$ then
 Output 'E'
 else Output 'F'

مثال على: بنية الاختيار المعشعبة
if / elseif elseif ... else



بنية الاختيار المعشعبة-2-

```
#include "stdafx.h"
#include<iostream>
using namespace std;
void main()
{
    int mark; cout<<"enter mark:";
    cin>> mark;
    if(mark >=95) cout<<"A\n";
    else if(mark >=90) cout<<"B\n";
        else if(mark >= 80) cout<<"C\n";
            else if(mark >= 70) cout<<"D\n";
                else if(mark >= 60) cout<<"E\n";
                    else cout<<"F"<<endl;
}
```

If student 's mark is greater than or equal to 95

Print "A"

else

If student 's mark is greater than or equal to 90

Print "B"

else

If student 's mark is greater than or equal to 80

Print "C"

else

If student 's mark is greater than or equal to 70

Print "D"

else

If student 's mark is greater than or equal to 60

Print "E"

else

Print "F"

إن نص التمرين معبر عن الخوارزمية
المحكية وهنا يوجد تقييم آخر للمراتب.

مثال على: بنية الاختيار المشعبة
if / elseif ... else



بنية الاختيار المشعبة-2-

```
if ( grade >= 90 ) // 90 and above
    cout << "A";
else if ( grade >= 80 ) // 80-89
    cout << "B";
else if ( grade >= 70 ) // 70-79
    cout << "C";
else if ( grade >= 60 ) // 60-69
    cout << "D";
else // less than 60
    cout << "F";
```

ملاحظة:

هذا النص صحيح في C++
ولكن عند Python غير
محترم لقواعد المسافات مما
يجعلنا أكثر ميل للنص
السابق

[1]

(1) switch

The multiple selection structure .switch -4-3-3 بنية متعدد الاختيار switch

الصيغة العامة (تكافئ فقرة متعدد الخيارات السابقة مؤلفة من شرط وجسم)

```
switch (expression)
```

```
{
```

```
    case value1: statement1; break;
```

```
    case value2 : statement2; break;
```

```
    case value3: statement3; break;
```

```
    default : statement_n ;
```

```
}
```

switch : كلمة مفتاحية (محمولة) يليها قوسين صغيرين يوضع بينهما قيمة أو **expression** تعبير يؤول لقيمه ستمرر وسيتم اختبارها.

من جسم التعليمة إلى مابعد { بفضل تعليمة **break** أي نتجاهل الباقي، وإلا ستختبر إن كانت تساوي **value2** تنفذ **statement2** وتخرج وهكذا وإذا لم تتطابق مع أي من **values** تنفذ الحالة الافتراضية **default** وهي اختيارية قد لا تكون موجوده دائماً، ووجود **break** فيها اختياري ولن يؤدي عمل مميز.

مثال 13: بنية متعدد الاختيار switch 2

```
#include "stdafx.h"
#include<iostream>
using namespace std;
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;
        //default : cout << "error ?enter an operation\n "
    }
;
}
```

ليكن المطلوب إدخال أحد رموز العمليات الحسابية الأساسية وإجراء هذه العملية على عددين مدخلين حيث تستخدم تعليمة switch وكتابة الخوارزمية.

Text Algorithm (Pseudo code)

1. Start
2. Input x ,y, ch
3. switch (ch)
 - case ch='+' , output x+y , break .
 - case ch='-' , output x-y , break .
 - case ch='*' , output x*y , break .
 - case ch='/' , output x/y , break .
 - case ch='%' , output x%y , break .
4. end

ملاحظة: 1- يمكن إضافة الحالة الافتراضية كسطر أخير default .
2- عند انطباق عدد من الحالات بخيار واحد نكتب case 1 : case 2: statn

مثال 14: بنية متعدد الاختيار switch (3)

```
#include "stdafx.h"
#include<iostream>
using namespace std;
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;
}
}
```

ليكن المطلوب طباعة red إذا كان المحرف المدخل r أو R،
ويطبع 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.

ملاحظة: إذا كانت الثوابت constant أعداداً
صحيحة (ليست محارفاً) عندئذ لا توضع هذه
الثوابت ضمن فواصل علوية.

تمرين:
ليكن المطلوب إدخال رقم شهر من أشهر
السنة ليتم طباعة أسم هذا الشهر

بعض الأسئلة المطلوب

بعض الأسئلة المطلوب الإجابة عن التالي:

- 1 - المطلوب مثال عن المخطط التدفقي بخيار واحد وبخيارين ومتعدد الخيارات.
- 2 - مقارنة ما بين `switch & else ... else if`
- 3- ليكن المطلوب كتابة الخوارزمية باستخدام اللغة الطبيعية وسم المخطط الانسيابي وكتابة الكود البرمجي للمعادلة التالية:

$$y = \frac{2x + 3m}{2x - 4}$$

- 4- ليكن المطلوب إدخال درجة طالب g وإذا كانت $g \leq 60$ اطبع `passed` و `good` وإلا اطبع `failed` و `good luck` كل كلمة تطبع على سطر منفرد مستخدم الشكل المختصر للتعليمة `if`.
- 5- تتبع المثال 11 للقيم التالية: $a=1, b=4, c=3$.
- 6- جدد مجال تحقق لعبارة إن أمكن ومبرر السبب . $(a < 0) \&\& (a > 100)$