



المعلوماتية كلية الهندسة

مدخل إلى الخوارزميات والبرمجة

**Introduction to Algorithms and Programming**

ا. د. علي عمران سليمان

محاضرات الأسبوع الثامنة

**الفصل الثاني 2022-2023**

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

المحاضرة من المراجع :

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

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

أعطيت كل من min, max قيمة ابتدائية = n  
واستخدمت القيمة المدخلة الأولى كقيمة ابتدائية.

```
void main()
{
int n,min,max; cout<<"enter integers n= ";
cout<<" terminate input with n=0:\n";
cin>>n;
//min=max=n ; for( ; n!=0 ; )
for(min=max=n ; n!=0 ; )
{ if(n<min) min=n;
else if(n>max) max=n;
cout<<" terminate input with n=0:\n";
cin>>n;
}
cout<<"min="<<min;
cout<<" "<<"max="<<max<<endl;
}
```

يطلب كتابة  
الخوارزمية  
النصية  
ورسم  
المخطط  
الانسيابي  
للتمرين

مثال 26: انتهاء الحلقة بسبب قيمة مدخلة،  
ليكن المطلوب إدخال متتالية من الأعداد  
الصحيحة تنتهي بالعدد صفر ويطبع أكبرها  
وأصغرها بالاعتماد البنية for. و while

```
void main()
{
int n,min,max; cout<<"enter integers n= ";
cout<<" terminate input with n=0:\n";
cin>>n; min=max=n;
while(n!=0)
{ if(n<min) min=n;
else if(n>max) max=n;
cout<<" terminate input with n=0:\n";
cin>>n; //input for the next loop
}
cout<<"min="<<min;
cout<<" "<<"max="<<max<<endl;
}
```

## مثال 27 حساب الفائدة المركبة حلقة معروفة عدد مرات التكرار

حيث  $a = p(1 + r)^n$  هو المبلغ الأساس،  $r$  هو معدل الفائدة السنوية مثلاً 0.05 أي 5٪،  $n$  هو عدد السنوات،  $a$  هو المبلغ المودع (المدور) في نهاية العام.

مثال 27: وضع مستثمر؟ مبلغ account في البنك لعدد من السنوات years، ما هي قيمة هذا المبلغ بعد كل سنة منها علماً الفائدة السنوية هي 5 % يطلب طباعة المبلغ بعد كل سنة وذلك لمدة السنوات years المدخلة

```
int main()
{double account; int years;
cout<<"Enter account : ";cin>>account;
cout<<"Enter number of years : "; cin>>years;
for(int year = 1; year <= years ; year++)
{ double amount = account * pow(1.0 + 0.05, year);
cout<<"\n amount after "<<year<<" years = "<<amount<<endl;
}
system ("pause");return 0; }
```

## حلقة معروفة مجال عملها وبخطوات مختلفة

مثال 28 ليكن المطلوب برنامج يطبق على الاعداد من 1 إلى 100: ا- يطبع مجموعها . ب- طباعة مجموع الأعداد الزوجية منها. ت- طباعة مجموع الأعداد الزوجية التي تقبل القسمة على العدد 5.

```
void main()
{
int sum=0,sum1=0, sum2=0;
cout<<"the sum of the num from 1 to 100 is:";
for(int number=1;number<=100;number +=1)
sum += number; cout <<sum<<endl;
cout<<"the sum of the even number between 1
and 100 is:";
for(int num2=2;num2<=100; num2 +=2)
sum2 += num2; cout<<sum2<<endl;
```

```
cout<<"the sum of the even number" <<"
between 1 and 100 that divide 5 is:";
for(int number1=2;number1<=100;number1 +=2)
if(number1%5 == 0)
sum1+=number1;

cout<<sum1<<endl; }
```

ملاحظته: لم نلتزم بالمسافات البادئة التي نوضي بها دائماً لعدم توفر المساحات على الشريحة

## حلقة معروفة مجال عملها



مثال 28 برنامج لأعداد factorial  
مثال 29 برنامج لأعداد Fibonacci

```
void main(void)
{
    int n, fibn;
    cout<<"Input n<2 to end\n" ; cin>>n;
    int i=2,fib0=0,fib1=1;
    while( i<=n){ fibn=fib0+fib1; i++;
        int tem=fib0;fib0=fib1;fib1=fib0+tem;
    }
    cout<<"the fibonacci of "<< n <<" = "<<fibn<<endl;
    system ("pause");
}
```

```
void main(void)
{
    int a, fac=1;
    cout<<"Input a 0 to end\n" ;
    cin>>a;
    while( a >0)
        { fac=fac*a; a--;}
    cout<<"the factorial of a= "<<fac;
    system ("pause");
}
```

صفوف التخزين: auto, register, static, mutable, extern، سنتركها لبحث التوابع

قواعد مجالات الرؤيا: يتحدد مجال الرؤية المتعلق بمعرف ما بجزء البرنامج الذي نستطيع ضمنه التعامل مع هذا المعرف، فمثلاً: عندما نصرح عن متحول محلي ضمن كتلة فيمكن التعامل مع هذا المتحول فقط ضمن الكتلة والكتل بداخلها والتي لم تقم بتعريفه ثانياً. ويوجد أربعة مجالات رؤية ممكنة للمعرف هي:

1. مجال الرؤية على مستوى التابع function scope.

2. مجال الرؤية على مستوى الملف file scope.

3. مجال الرؤية على مستوى الكتلة block scope.

4. مجال الرؤية على مستوى نماذج التوابع scope function-prototype.

سندرس الآن على مستوى الكتل فقط تاركين الباقي لبحث التوابع:

تتمتع المعارف المصرح عنها ضمن كتلة ما بمجال الرؤية على مستوى هذه الكتلة، ويبدأ من النقطة التي وردت فيها التصريح عن المعروف وحتى نهاية الكتلة المحددة بالقوس اليميني المغلق له (}). والمعارف المحلية: هي المصرح عنها في بداية التابع بالإضافة إلى معارف وسطاء هذا التابع (الموجوده ضمن القوسين الدائرين عند كتابته) والمعتبره أيضاً كمتحولات محلية له.

المعرفات في رأس حلقة for هي خاصة بها  
وتعامل معاملة معرفات الكتلة block



## قواعد مجالات الرؤية scope rules

```
int main()
{int i=7; cout<<"\ni= "<<i;
{cout<<"\ni= "<<i; int i=15; cout<<"\ni= "<<i;
{cout<<"\ni= "<<i; int i=0; cout<<"\ni= "<<i;}
cout<<"\ni= "<<i;}

for ( int i = 1 ; i<=4; i++)
{cout <<"\ni= "<< i ;}
cout <<"\ni= "<< i<<endl;
  system ("pause");
return 0;}
```

i= 7  
i= 7  
i= 15  
i= 15  
i= 0  
i= 15  
i= 1  
i= 2  
i= 3  
i= 4  
i= 7

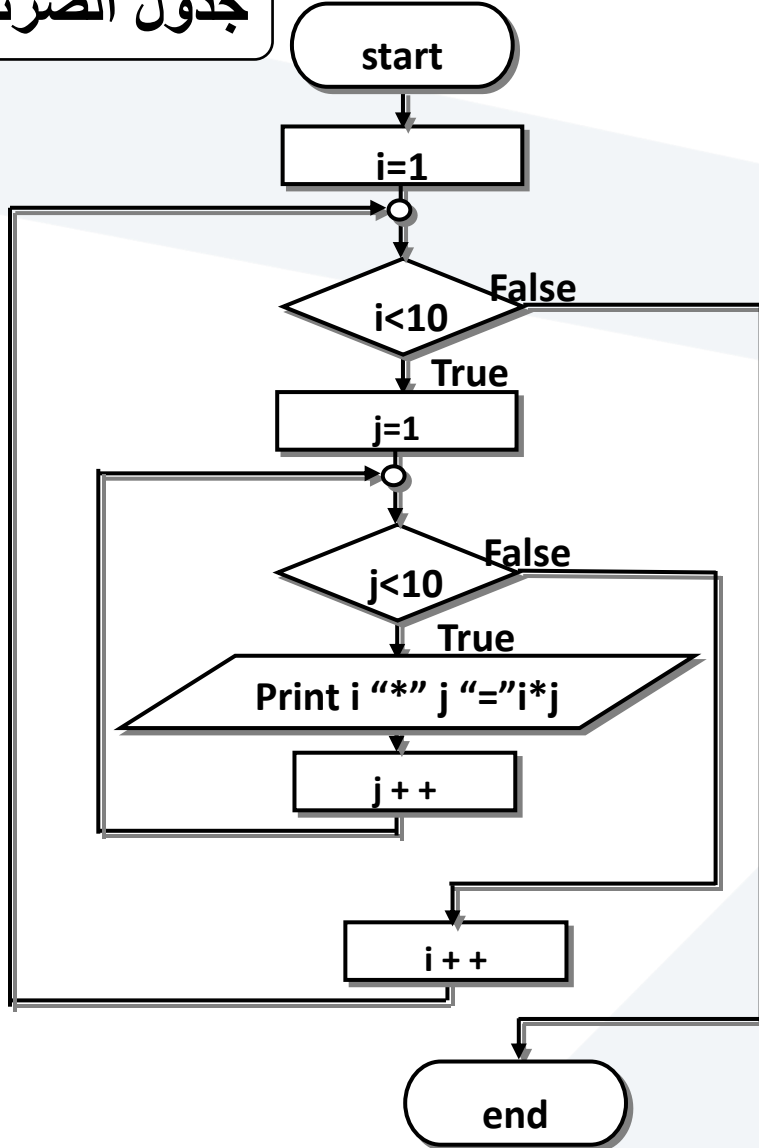
الكتلة يمكن أن تحتوي بداخلها كتل أخرى  
وعند التصريح عن المعرف ضمن الكتلة  
الخارجية يكون مرئي عليها وعلى كل  
الكتل ضمنها، إلا إذا عرف من جديد ضمن  
الكتلة الداخلية فيصبح المعرف الخارجي  
مخفي عن الكتلة الداخلية التي عرف بها  
حتى نهايتها ليعود للظهور من جديد،  
أي أثناء التنفيذ المتعلق بالكتل الداخلية  
فإنها ترى فقط المعرفات المحلية الخاصة  
بها إن عرفت ولا ترى المعرفات التي  
تحمل نفس أسماء معرفاتها والموجودة  
في الكتل التي احتوتها.

المعرفات في رأس حلقة for هي خاصة بها وتعامل معاملة معرفات الكتلة block



### 8-3-3- الحلقات المتداخلة The nested loops

#### جدول الضرب



عندما يتطلب الأمر إجراء حلقة لكل قيمة من قيم حلقة أخرى يكون ضرورياً وضع حلقة ضمن حلقة أخرى، وسيتم إنهاء قيم الحلقة الداخلية لكل قيمة من قيم الحلقة الخارجية. نستدعي `#include <iomanip>` من أجل بناء الجداول بـ 5 فئات نستخدم التابع `setw(5)` ومن أجل تعبئة هذه الفراغات بمحارف مثل 0 نستخدم التابع `setfill('0')`

مثال 31 جدول الضرب للأعداد من 1 حتى 9 .

```

void main()
{for(int i=1;i<10;i++)
  for(int j=1;j<10;j++)
    cout<<i<<"*"<<j<<"="<<i*j<<endl;
}
// #include <iomanip> for setw(5), setfill('0')
//cout<<setw(8)<<i<<"*"<<j<<"="<<i*j;if(j==9)cout<<endl;}
//{ for {if(j<=i)cout<<setw(5)<<i*j;}cout<<endl;}
  
```

## خوارزمية اقليدس لحساب القاسم المشترك الأعظم لعددين

مثال 32: ليكن المطلوب حساب القاسم المشترك الأعظم لعددين باستخدام خوارزمية اقليدس

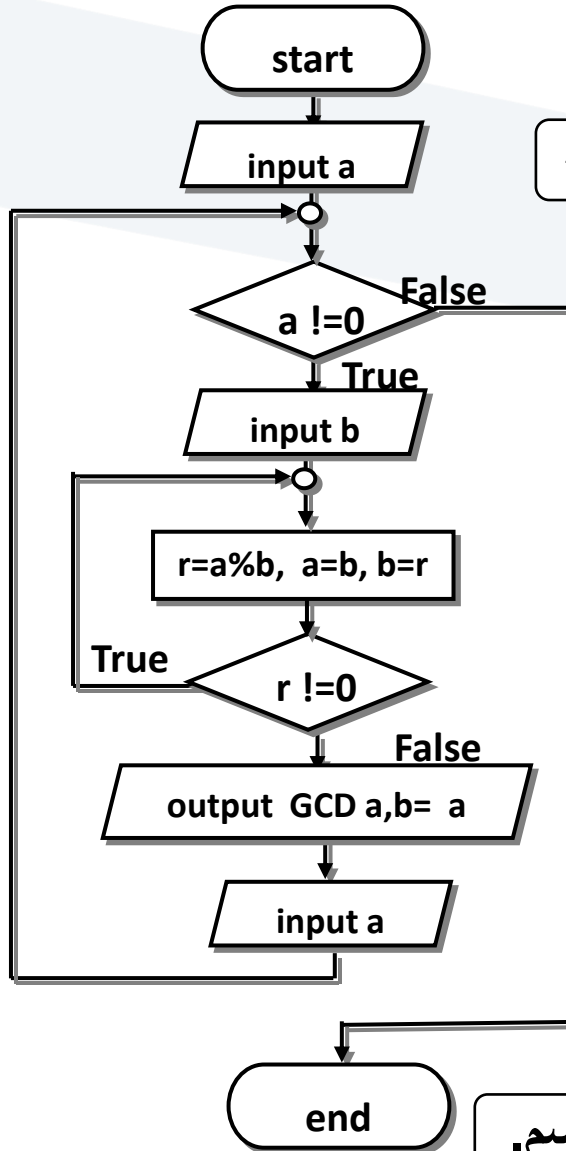
```
void main()
{
    int a, b, r, temp;
    cout<<" input a, b ";
    cin>>a>>b;
    if(b!=0 && a!=0)
    { if(a<b){temp = a; a=b; b=temp;}
    do
        { r=a%b; a=b; b=r; }
    while(r!=0);
    cout<<" GCD a,b= "<<a<<endl;
    }
    else cout<<"a=0 || b=0 \n";
    system ("pause");
}
```

يطلب من الطالب :

1. كتابة الخوارزمية النصية.
2. رسم المخطط الانسيابي.

a= 30	b= 8		r= 6
a= 8	b= 6	? 6!=0 ✓	r= 2
a= 6	b= 2	? 2!=0 ✓	r= 0
a= 2	b= 0	? 0!=0 x	

مثال 33: ليكن المطلوب تطوير البرنامج السابق بحيث يستخدم حلقة ثانية لجعل البرنامج يتكرر حسب الرغبة



```

void main()
{
  int a, b, r; cin>>a;
  while( a !=0)
  {
    cin>>b;
    if(a<b){temp = a; a=b; b=temp;}
    do
    { r=a%b; a=b; b=r; }
    while(r!=0) ;
    cout<<" GCD a,b= "<<a;
    cout<<"Inp. a 0 to end" ;
    cin>>a;
  }
}
  
```

Suppose we have two positive inte, and the first is greater than the second, otherwise this is handled programmatically.

**Start**

**Input a**

**While a value not equal to zero**

**Input b**

{ if(a<b){temp = a; a=b; b=temp;}

do r=a%b; a=b; b=r;

while r value not equal to zero

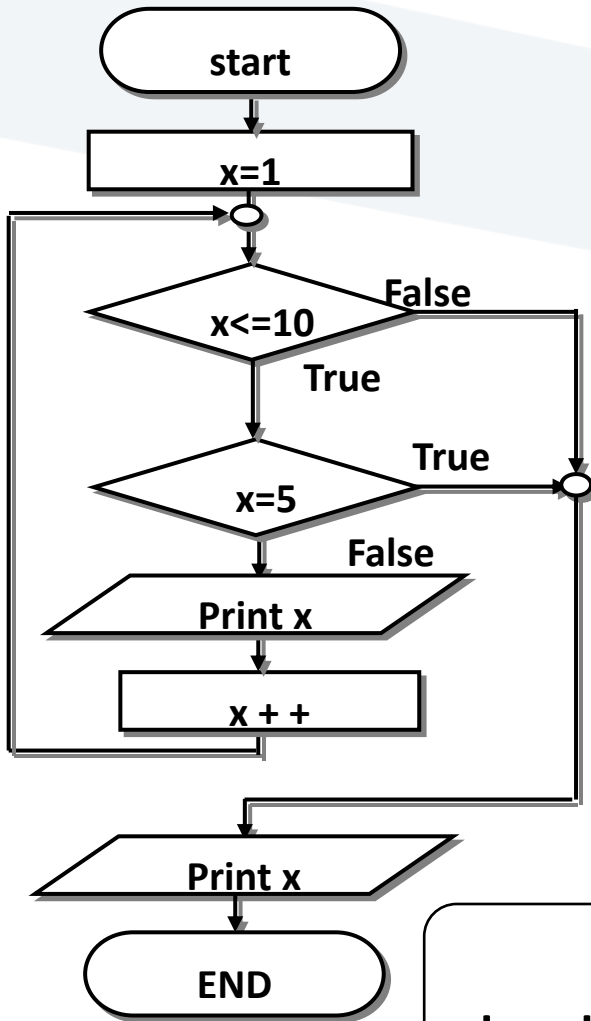
output GCD a,b= a

**Input a**

**End**

يتوقف التمرين عن التنفيذ عند إدخال العدد a=0 سنستخدم حلقة تنتظر قيمة مدخلة sentil وحلقة لحساب القاسم.

## 4-3- التعليمتان break, continue



```

void main()
{ int x;
  for( x=1;x<=10;x++)
  {
    if(x==5)break;
    cout<<x<<" ";
  }
  cout<<endl<<"broke out
of loop at x=="<<x<<endl;
}
  
```

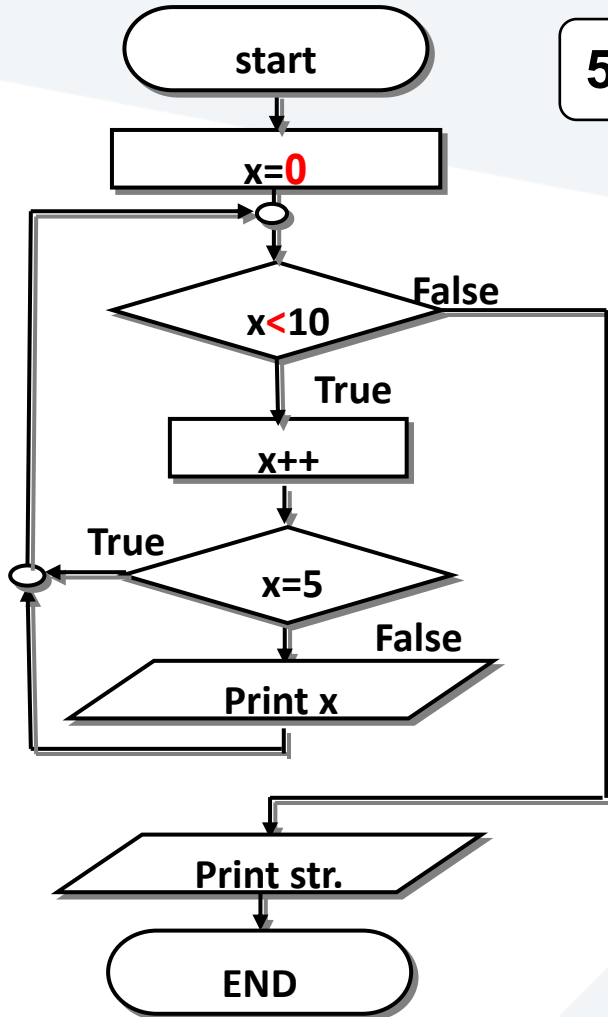
تستخدم التعليمتان لتغيير مجرى التحكم ضمن البرنامج.  
أ- التعليمة **break**:  
تسبب تعليمة **break** عند تنفيذها مع أحد البنى التكرارية **while, for, do\while** ومع بنية الاختيار **switch** الخروج مباشرة من جسم البنية ويتابع البرنامج بعدها تنفيذها مع أول تعليمة تليها مباشرة.

مثال 34 ليكون المطلوب قطع تنفيذ الحلقة عند القيمة 5

نستخدم تعليمة **break** كذلك للخروج من حلقة لانهاية، فالبنى **while(1){statements ; ...};** **for( ; ; ){statements; .. };** **do{statements; ...}while(1);** لاتنتهي بفعل شرطها بل مع شرط ثم **break**

## 4-3- التعليمتان break, continue

مثال 35: ليكن المطلوب طباعة الاعداد من 1 إلى 10 بدون 5



```

void main()
{
int x=0;
While(x<10)
{ x++; if(x==5)continue;
cout<<x<<" ";
}
cout <<"\n used continue to
skip printing the value 5\n" ;
}
  
```

ب- التعليمة continue .  
تسبب continue عند تنفيذها مع أحد البنى while, do\while, for، إلى تجاوز ما تبقى من جسم البنية التكرارية والمتابعة مع المرور التالي للبنية.  
يتم في البنيتين do\while, while القيام بالتحقق من صحة شرط استمرار التكرار مباشرة بعد تنفيذ تعليمة continue.

البنية for فيتم القيام بعملية التحديث أولاً ثم يبدأ بعدها تقييم شرط استمرار الحلقة.

البنيتين do\while, while يجب تحديث العداد قبل تعليمة continue وإلا سندخل بحلقة لانتهائية.

# بعض الأسئلة المطلوب

## بعض الأسئلة المطلوب الإجابة عن التالي:

1. ليكن المطلوب برنامج يقوم بإيجاد ناتج ضرب عددين صحيحين  $a$  ,  $b$  بدون استخدام معامل الضرب، حيث  $a$  ،  $b$  عددين موجبين باستخدام الحلقات الثلاث.
2. ليكن المطلوب برنامج يقوم بإيجاد  $n!$  باستخدام `while(1)` وتعليمة `break`.
3. المطلوب تصميم برنامج يؤدي المهام التالية:  
يتم اختبار طفل في عمليات الجمع والطرح والضرب والقسمة ويقوم بإدخال عددين من النوع الصحيح ومن ثم يطالب بإدخال العملية ويطالب بإدخال الناتج ويختبر الناتج ليعطي عبارة شكر مثل " ممتاز شاطر جداً " إذا كان الجواب صحيح وعبارة تشجيع للمتابعة إذا كان الجواب خطأ " حاول مره ثانية " ويقبل منه جوابين خطأ وفي المرة الثالثة يعطيه الجواب الصحيح وعبارة " لا بأس حظ أوفر في العملية التالية " ومن ثم يسأل عن الرغبة في المتابعة في البرنامج " هل تريد المزيد من المحاولات " وفي حال الرغبة يكرر ما سبق وفي حال الرغبة بالخروج من البرنامج يتم ذلك.
- 4- طور البرنامج السابق ليعطي إحصائية عن استثمار البرنامج نسب الصح أو نسب الخطأ في إجابته.