

```

#include<iostream>
using namespace std;
main()
{
1.int a,b;
2.cin>>a>>b;
if ( a>b )
3.cout<<" a>b ";
  if ( a>=b )
4.cout<<"\n a>=b ";
if ( a<b )
5.cout<<"\n a<b ";
if ( a<=b )
6.cout<<"\n a<=b ";
If ( a==b )
7.cout<<"\n a==b ";
if ( a!=b )
8.cout<<"\n a!=b ";}

```

توضيح الخطوات:

1. تنفذ أولاً خطوة رقم 1 هي تعريف للمتغيرات a , b . ثم تنفذ خطوة رقم 2 هي قراءة (ادخال) للمتغيرات التي ستدخل قيمها من قبل المستخدم الآن لنختبر أن ندخل قيم من شاشة التنفيذ ونرى النتائج
2. لو أدخلنا a=3,b=2 كما لاحظت قد تنفذ الخطوات رقم 3,4,8 لأن شروطها تحققت وذلك لأنه
3. الشرط التابع للخطوة رقم 3 هو أن يكون قيمة a اكبر من قيمة b وفعلاً أن قيمة a التي أدخلناها كانت 3 وقيمة b=2 لذلك نفذ الخطوة الثالثة لاحظ التحقق من الشرط

True (3>2) ← (a>b)

4. الشرط التابع للخطوة رقم 4 هو أن يكون قيمة a اكبر أو يساوي قيمة b وفعلاً أن قيمة a التي أدخلناها كانت 3 أي كانت اكبر b=2 لذلك نفذ الخطوة 4 * وشرط (=) يتحقق إذا كان الرقم اكبر منه أو يساويه ينفذ عبارة التابعة للشرط

True (3>2) ← (a>=b)

5. الشرط التابع للخطوة رقم 8 هو أن يكون قيمة a لا يساوي قيمة b وفعلاً أن قيمة a التي أدخلناها كانت 3 وقيمة b=2 أي كانت لا تساوي لذلك نفذ الخطوة رقم (8)

وبقية الخطوات لم تنفذ لأنها لم تتحقق شروطها

على سبيل المثال لماذا لم تنفذ الخطوة رقم (5) ؟ لأن شرطها أن يكون a اصغر من قيمة b وان هذا الشرط لا يتحقق لأن قيمة a اكبر من قيمة b \rightarrow False $(a < b) (3 < 2)$

نتيجة المقارنة هي false لذلك سوف لا ينفذ الخطوة التي تتبعه لأن الشرط لم يتحقق لو أدخلنا $a=3, b=6$ كما لاحظت قد تحققت الخطوات رقم (5,6,8) لأن شروطها تحققت وذلك لأنه

1. الشرط التابع للخطوة 5 هو أن يكون قيمة a اصغر من قيمة b وفعلا أن قيمة a التي أدخلناها كانت 3 وقيمة $b=6$ لذلك نفذ الخطوة رقم 5

$(a < b) (3 < 6) \rightarrow \text{True}$

الشرط التابع للخطوة 6 هو أن يكون قيمة a اصغر أو يساوي قيمة b وفعلا أن قيمة a التي أدخلناها كانت 3 أي كانت اصغر $b=6$ كانت 6

* وشرط (\geq) يتحقق إذا كان الرقم اصغر منه أو يساويه ينفذ عبارة التابعة للشرط

$(a \leq b) (3 \leq 6) \rightarrow \text{True}$

الشرط التابع للخطوة 8 هو أن يكون قيمة a لا يساوي قيمة b وفعلا أن قيمة a التي أدخلناها كانت 3 وقيمة $b=6$ أي كانت لا تساوي لذلك نفذ الخطوة 8

$(a \neq b) (3 \neq 6) \rightarrow \text{True}$

***وبقية الخطوات لم تنفذ لأنها لم تتحقق شروطها

لو أدخلنا ($a=3, b=3$) كما لاحظت قد تحققت الخطوات رقم (4,6,7) وذلك لأنه

الشرط التابع للخطوة 4 هو أن يكون قيمة a اكبر أو يساوي قيمة b وفعلا أن قيمة a التي أدخلناها كانت 3 وقيمة $b=3$ وكانت متساويتان $b=3$ كانت 4

$(a \geq b) (3 \geq 3) \rightarrow \text{True}$

الشرط التابع للخطوة 6 هو أن يكون قيمة a اصغر أو يساوي قيمة b وفعلا أن قيمة a التي أدخلناها كانت 3 أي كانت تساوي $b=3$ لذلك نفذ الخطوة 6

* وشرط (\geq) يتحقق إذا كان الرقم اصغر منه أو يساويه ينفذ عبارة التابعة للشرط

$(a \leq b) (3 \leq 3) \rightarrow \text{True}$

الشرط التابع للخطوة 7 هو أن يكون قيمة a تساوي قيمة b وفعلا أن قيمة a التي أدخلناها كانت 3 أي كانت تساوي $b=3$ كانت 7

$(a == b) (3 == 3) \rightarrow \text{True}$

```

#include<iostream>
Using namespace std;
main(){
int a;
cin>>a;
if((a>5)&&(a<100)){
    cout<<"the number in this range";
}

}

```

****** وبقية الخطوات لم تنفذ لأنها لم تتحقق شروطها

توضيح الخطوات :

1. خطوة رقم (1) عرفنا متغير من نوع integer لكي نحمله بالقيمة المدخلة
2. خطوة رقم (2) هي إدخال قيمة للمتغير من قبل المستخدم
3. خطوة رقم (3) هو عملية التحقق من الشرط
4. فلو ادخل المستخدم الرقم (50) فتكون المقارنة بشكل التالي:

$(20 > 5 \&\& 20 < 100) \text{ true } \&\& \text{true} = \text{true}$

وإذا كان شرط ال if هو true ينفذ statement اي مافي داخل اقواس ال if.

$(150 > 5 \&\& 150 < 100) \text{ true } \&\& \text{false} = \text{false}$

وإذا كان شرط ال if هو false لا ينفذ statement اي مافي داخل اقواس ال if.

```

#include<iostream>
Using namespace std;
main(){
1. int a=5;
2. int b=4;
if (b%2==1){
3. a=a+6;
4. b=b+4;
}
5. a=a+3;
6. b=b+2;
7. cout<<"a="<<a<<"\tb="<<b;
}

```

1. خطوة رقم (1) أصبح قيمة (a=5)
2. خطوة رقم (2) أصبح قيمة (b=4) وقيمة (a) بقيت ثابتة لم تتغير محتفظة بقيمتها في الخطوة السابقة (a=5)
3. قبل الخطوة رقم (3) يوجد شرط لتنفيذ ما بين قوسي العبارة الشرطية وهو أن يكون باقي قسمة قيمة المتغير b على (2) تساوي واحد أن يكون رقم فردي وان باقي قسمته على (2) هو صفر لذلك سوف لا ينفذ الخطوة رقم (3 و4) وينتقل إلى خطوة رقم (5) وتكون المقارنة بشكل التالي
 False (0==1) (4%2==1) (b%2==1)
4. خطوة رقم 5 أصبح قيمة a هي a=5+3=8 وقيمة b بقيت ثابتة لم تتغير b=4
5. خطوة رقم 6 أصبح قيمة b هي b=4+2=6 وقيمة a بقيت ثابتة لم تتغير a=8
 والنتيجة على شاشة التنفيذ هي b=6 a=8

```
#include<iostream>
Using namespace std;
main(){
1. int a;
2. cin>>a;
   If ( a>0 )
3. cout<<"is greater than 1 is positive";
   else
4. cout<<"is less than 1 is negative";
5. a=a+6;
6. }
```

تتبع خطوات البرنامج:-

1. خطوة رقم 1 تم تعريف متغير a من نوع integer
 2. خطوة رقم 2 يطلب من المستخدم إدخال قيمة للمتغير a من شاشة التنفيذ
 3. ألان نأتي إلى شرط التحقق
 إذا ادخل المستخدم إي رقم اكبر من صفر سوف يحقق شرط if و ينفذ السطر 3 ويطبع هذه الرسالة
 greater than 1 is positive
 ولا ينفذ السطر 4 لأنه يتحقق إذا لم ينفذ الشرط تم ينفذ السطر 5 تم السطر 6 إي يكون تسلسل تنفيذ خطوات البرنامج الكلية
- 1 2 3 5 6
- إذا ادخل المستخدم إي رقم اصغر من صفر سوف لا يحقق شرط if ولذلك ينتقل إلى else و ينفذ السطر 4 ويطبع هذه الرسالة في شاشة التنفيذ

is less than 1 is negative

تم ينفذ السطر 5 تم السطر 6 إي يكون تسلسل تنفيذ خطوات البرنامج الكلية
1 2 4 5 6

```
#include<iostream>
using namespace std;
main(){
    int input_Try;
    cin>>input_Try;
    if (input_Try >0 )
    {
        if (input_Try > 5)
            cout<<"the number is greater than 5";
        else
            cout<<" the number is less than 5";
    }
}
```

❖ مستخدما لغة السي بلص بلص اكتب برنامج يطلب من المستخدم ادخال رقم اذا كان الرقم موجب يجب ان يختبر الرقم اذا كان اكبر من خمسة اولا اذا كان اكبر من خمسة يطبع رساله انه اكبر من خمسة واذا لا يطبع رساله بانه اقل من خمسة (باستخدام العبارات المتداخلة)

وان عبارة if –else الداخليه لا تنفذ مطلقا ولا يمر عليها المترجم إلا بتحقق شرط العبارة الشرطية إلا وهي if (input_Try > 0) إي أن يكون الرقم المدخل input_Try اكبر من صفر إذا كان الرقم المدخل فعلا اكبر من صفر سوف يمر على عبارة if –else الداخليه وإما ينفذ الخطوة رقم 1 أو الخطوة رقم 2

وإذا لم يكن اكبر من صفر لا يمر أصلا على عبارة if –else الداخليه ويتجه نحو الخطوة 3 لينفذها لان عبارة if –else الداخليه واقعة ضمن عبارة الشرطية if (input_Try > 0) وبما أن شرطها لم يتحقق لذلك سوف لا ينفذ ما هو موجود داخل قوسيتها إذا جاء لدينا أكثر من if وأكثر من else فكيف نعرف أن هذه if –else تنفي العبارة الشرطية if الأولى أم الثانية أم غيرهما..؟ بكل سهولة كل else تنفي اقرب عبارة if الشرطية عليها.

مثال: برنامج تدخل رقم ويبين لكل هل الرقم يقبل القسمة على 3 أم على 5 أم على 7 أم غير ذلك
ويطبع رسالة في كل حالة في حال إذا قبل القسمة على 7 يجمع مع الرقم المدخل قيمة 2 ويطبعه

```
#include<iostream>
Using namespace std;
main()
{
1.int number_enter;
2.cin>> number_enter;
if ( number_enter % 3==0)
3.cout<<"Accept Devide to 3" ;
else if (number_enter % 5==0)
4. cout<<" Accept Devide to 5" ;
else if (number_enter % 7==0)
{
5.number_enter= number_enter+2;
6.    cout<<" Accept Devide to 7=" << number_enter;
}
else
7.    cout<<"Donot True any of conditions" ;

}
```

تتبع خطوات البرنامج مع التوضيح :

1. خطوة رقم 1 تنفذ بشكل اعتيادي
2. خطوة رقم 2 تنفذ بشكل اعتيادي ويطلب من المستخدم إدخال قيمة للمتغير number_enter من شاشة التنفيذ لنفرض أننا أدخلنا number_enter مرة (6 و14 و15 و20 و 19)
لنتتبع القيم المدخلة
number_enter=6 إذا كان
سوف ينفذ الخطوة رقم 3 وينتهي البرنامج حسب شرح عن الدالة if—else if إذا تحقق احد الشروط في احد العبارات تنفذ ما في داخله وتتجه إلى ما بعد else النهائية لتنفذه وبما انه لا يوجد شي بعد else يوجد فقط قوس نهاية البرنامج لذلك ستنهي البرنامج طابعا للرسالة التالية تم وينتهي البرنامج Accept Devide to 3
number_enter=14 إذا كان
سوف ينفذ الخطوة رقم 5 وخطوة رقم 6 طابعا للرسالة التالية تم وينتهي البرنامج

Accept Devide to 7=9

number_enter=15 إذا كان

سوف ينفذ الخطوة رقم 3 ولا ينفذ خطوة رقم 4 رغم قبوله القسمه على 5 لأنه برمجيا عندما يبدأ البرنامج بالتحقق من الشروط يبدأ بالتحقق بالتسلسل خطوة بخطوة حسب الترتيب ونحن واضعين شرط قبول القسمه على ثلاثة قبل شرط قبول القسمه على خمسة وعندما يمر البرنامج على الشرط الأول وحققه لا يمر على الشرط الثاني

بل يتجه إلى ما بعد (.else) لو كنا واضعين شرط قبول القسمه على خمسة قبل شرط قبول القسمه على ثلاثة لتنفيذ خطوة رقم 4. لذلك سيطبع البرنامج الآن هذه الرسالة Accept Devide to 3

number_enter=20 إذا كان

سوف ينفذ الخطوة رقم 4 طابعا للرسالة التالية تم وينتهي البرنامج

Accept Devide to 5

number_enter=19 إذا كان

سوف لن يحقق أي من الشروط السابقة لذلك سيتجه إلى else ينفذ الخطوة رقم 7 طابعا للرسالة التالية تم وينتهي البرنامج

Donot True any of conditions

خطوة رقم 5 و6 وضعت بين قوسين للدلالة على أنهما تابعتان لعبارة else

❖ مستخدما لغة اكتب برنامج ولنفترض لدينا المعومات التالية عن هؤلاء الأشخاص نريد مجرد كتابة أول حرف من اسم الشخص يعطيك المعلومات الكاملة عنه ؟

```

#include<iostream>
using namespace std;
main()
{ 1.char index_of_Name;
2.Cout<<"Enter First Chat of Student name: ";
3.Cin>> index_of_Name ;
4.switch( index_of_Name ) {
5.case 'A':
6.Cout<<" his names Ali kammel,20 Year old, third stage " ;
7.break;
8.case 'S':
9.Cout<<" his names Salem kammel,18 Year old, third stage " ;
10.break;
11.case 'H':
12.Cout<<" his names Hussien Ahmmed Taleb,21 Year old, third stage
eng.Computer " ;
13.break;
14.default:
.15Cout<<"You Not have saved names in this index ; " }
}

```

1. خطوة رقم 1 عرفنا متغير اسمه index_of_Name من نوع حرفي
2. خطوة رقم 2 طبع رسالة للمستخدم تطلب منه إدخال أول حرف من اسم الشخص هذه الرسائل مهمة جدا لواجه برنامجك حتى يعلم المستخدم كيف يتعامل مع برنامجك ما هو المطلوب منه وماذا يدخل فدائما حاول أن يكون برنامجك واضح للمستخدم بهذه الرسائل
3. خطوة رقم 3 قمنا بقراءة حرف من شاشة التنفيذ , خطوة 4 أدخلنا الحرف الذي ادخله المستخدم في قائمة الخيارات وهي switch
4. خطوة رقم 5 تحوي حالة شرطية لأول حرف من اسم الشخص Ail وتلاحظ الأحرف في المقارنة توضع بين علامة تنصيصية واحدة من الجانبين هكذا 'A'
5. خطوة رقم 6 تطبع سجل هذا الشخص إذا كان هو المطلوب , وخطوة رقم 7 توقف ال case لهذه الحالة وبقية الخطوات نفس الشيء

❖ مثال: لو أردنا عداد يعد من (0) إلى (4) أي يطبع على شاشة التنفيذ من (0 1 2 3 4) ؟.


```
#include<iostream>
Using namespace std;
main()
{
int i;
for (i=0; i<5; i++)
cout<<i<<"\t";
}
```

توضيح الخطوات :

1. خطوة رقم 1 عرفنا متغير i من نوع integer
2. خطوة رقم 2 هي عبارة تكرارية بما إننا نريد العد من الصفر فأعطينا القيم البدائية صفر والعد يصل إلى 4 معناه شرط التوقف أن يكون اقل من خمسة بما إننا نريد أن يعد خمس عدات بين صفر والأربعة لذلك يجب أن يكون مقدار الزيادة واحد ليعد 1,2,3,4*0). لان لو جعلنا مقدار الزيادة 2 مثلا فسيعد العداد ثلاث مرات وبشكل التالي(0,2,4) لذلك يجب التركيز في هذه المواضيع جيدا

❖ مثال: برنامج يدخل المستخدم درجات 10 مواد وتقوم بحساب المعدل له

```
#include<iostream>
Using namespace std;
main()
{
int n,aveg,input_number;
int sum=0;
for (n=0; n<10; n++) {
cin>> input_number;
sum += input_number ;
}
aveg=sum/10;
cout<<" aveg ="<< aveg ;
}
```

1. خطوة رقم 1 عرفنا متغير n ليكون عداد للعبارة التكرارية ومتغير حساب المعدل ومتغير إدخال الدرجات

2. خطوة رقم 2 عرفنا متغير sum من نوع integer وأعطيناه قيمة بدائية وهي صفر لأننا سنجمع والنضير الجمعي هو صفر
 3. خطوة رقم 3 هي جملة تكرارية مطلوب منها تكرار الخطوة رقم 4 وخطوة رقم 5 عشر مرات وبما انه مطلوب أن يكرر أكثر من سطر برمجي يجب وضعهما بين قوسي العبارة التكرارية
 4. خطوة رقم 4 هي دالة إدخال عند كل عدة للجملة التكرارية يطلب من المستخدم إدخال درجة جديدة
 5. خطوة رقم 5 يقوم بجمع الدرجات المدخلة عند كل إدخال يجمع الدرجة الجديدة مع ناتج جمع الدرجات السابقة
 6. خطوة رقم 6 يقوم بحساب المعدل حيث يقسم ناتج جمع الدرجات المدخلة على عشرة
 7. خطوة رقم 7 يقوم بطباعة المعدل
- عدد مرات تنفيذ الخطوات: تنفذ خطوة رقم 1 و 2 مرة واحدة فقط. وتنفذ خطوة رقم 3 و 4 و 5 عشر مرات وتنفذ خطوة رقم 6 و 7 مرة واحدة
- مثال: برنامج لإيجاد مفاكوك الإعداد.؟**

```
#include<iostream>
Using namespace std;
main()
{
int i,number;
int fact=1;
cin>> number ;
for(i=1;i<= number ;i++)
fact = fact *i;
cout<<"factorial= "<< fact ;
}
```

تحليل: المفاكوك هو عملية ضرب العدد بالإعداد التي هي اقل منه وصولاً إلى الواحد (مثلاً مفاكوك 6 هو $6! = 6 * 5 * 4 * 3 * 2 * 1$. إي أن $n! = n * (n - 1) * \dots * 1$ وبرمجياً لحل هذا السؤال نكون عدد يبدأ من الواحد وينتهي بالرقم المدخل ونضرب قيم العداد واحدة بالأخرى إلى النهاية نحصل على المفاكوك

توضيح خطوات :

1. خطوة رقم 1 تم تعريف متغير i كعداد لعبارة التكرار ومتغير آخر يحمل قيمة الرقم المراد إيجاد مفاكوكه
2. خطوة رقم 2 تم تعريف متغير fact كنضير ضربي يضرب في أجزاء الرقم المراد إيجاد مفاكوكه
3. خطوة رقم 3 يطلب من المستخدم إدخال الرقم المراد إيجاد مفاكوكه

4. خطوة رقم 4 عداد يعد من الواحد وحتى الرقم الذي تم إدخاله كان يكون إننا أدخلنا رقم 5 سيعد 1, 5, 4, 3, 2

5. خطوة رقم 5 هنا نضرب كل قيمة جديدة يعدها العداد بالقيم السابقة كان مثلاً إننا أدخلنا رقم 3 نريد إيجاد مفكوكه سيعضرب أولاً واحد في قيمة fact لتصبح قيمته $fact = 1 * 1 = 1$ ثم سيعضرب 2 في قيمة 1 الجديدة لتصبح قيمته $fact = 1 * 2 = 2$ ثم سيعضرب 3 في قيمة 2 الجديدة لتصبح قيمته $fact = 2 * 3 = 6$

تحليل : لو نركز في السؤال نراه يريد جدول ضرب وجدول الضرب مكون من ضرب رقمين فعلى سبيل المثال

جدول ضرب 1 يضرب رقم واحد بالأرقام من واحد إلى العشرة وجدول ضرب 2 يضرب الاثنان بالأرقام من واحد إلى عشرة ويستمر.....! أي إننا سنحتاج إلى عبارتين تكراريتين عبارة خارجية تخص جدول ضرب الرقم وأرقام داخلية تضرب هذه الرقم بالأرقام من واحد إلى عشرة.

```
#include<iostream>
using namespace std;
main()
{
int i,j;
for (i=1; i<=10; i++)
{
cout<<"Multiply tabel for("<<i<<").\n-----\n";
for (j=1; j<=10; j++)
cout<<j<<" * " <<i<<"="<<i*j<<"\n";
cout<<"\n";}
}
```

توضيح الخطوات :

1. خطوة رقم 1 تم تعريف متغيرين المتغير i هو جدول ضرب الرقم كان يكون جدول ضرب ثلاثة مثلاً والمتغير j هو يضرب بالمتغير الأول الأرقام من واحد إلى العشرة
2. خطوة رقم 2 هي عبارة تكرارية تعد من 1 إلى 10 وفي كل عدة لها تتكرر الخطوات رقم 3 و4 و5 و6 ولاحظ وضعنا هذه الخطوات بين قوسين لأنها أكثر من خطوة ونريد أن نتبعها لهذه العبارة التكرارية لذلك يجب وضعها بين قوسين
3. خطوة رقم 3 هي رسالة تظهر بداية كل جدول تبين للمستخدم انه هذا الجدول هو للرقم المعين وهي بضبط الرسالة الموجودة في شاشة التنفيذ في الأسفل ومؤشر عليها ب " جملة الطباعة الخطوة 3 "

وكما تلاحظ أن في كل عدة للعبارة التكرارية الخطوة رقم 2 تتكرر الخطوة رقم 3 مرة واحدة لتظهر بداية كل جدول

4. خطوة رقم 4 هي عبارة تكرارية داخلية وفي كل عدة للعبارة التكرارية في الخطوة رقم 2 تعمل هذه العبارة بالعد من 1 إلى 10 لكي تضرب كل رقم من الأرقام التي تعدها الخطوة رقم 2 بالأرقام من 1 إلى 10 وهذه العبارة التكرارية يتبعها سطر واحد فقط لعدم وجود أقواس

5. خطوة رقم 5 هي عملية طباعة كل قيمة من التي تعدها العبارة التكرارية في الخطوة رقم 2 بالأرقام من 1 إلى 10 التي تعدها العبارة التكرارية في خطوة رقم 4 وهذه جملة الطباعة هي تابعة لهذه العبارة التكرارية وما تنتجه هذه جملة الطباعة شاهده بالرسالة الموجودة في شاشة التنفيذ في الأسفل ومؤشر عليها ب " جملة الطباعة الخطوة 5 "

6. خطوة رقم 6 هي عملية طباعة سطر جديد بعد كل جدول ضرب ولاحظها هي تابعة للعبارة التكرارية في خطوة رقم 2 وما تنتجه هذه جملة الطباعة شاهده بالرسالة الموجودة في شاشة التنفيذ في الأسفل ومؤشر عليها ب " جملة الطباعة الخطوة 6 "

7. كيف نعرف أن الخطوة رقم 6 هي تابعة للخطوة رقم 2 وليس للخطوة رقم 4 ..؟

8. كما تلاحظ أن خطوة رقم 2 تحصر بين قوسيه خطوات رقم 3 و4 و5 و6 ونلاحظ أن العبارة التكرارية في الخطوة رقم 4 تتبعها فقط سطر واحد لأننا لم نضع أقواس خلفها أي إننا نقصد فقط السطر الذي يليها هو تابع لها فتبقى الخطوة رقم 6 وبما أنها داخل قوسين الخطوة رقم 2 فهي تابعة لها.

مثال: يطبع الشكل التالي في شاشة التنفيذ .؟

```
1
12
123
1234
12345
```

```
#include<iostream>
Using namespace std;
main()
{
int i,j;
for (i=2; i<7; i++)
{
for (j=1; j<i; j++)
```

```
cout<<j ;
cout<<"\n" ;}
}
```

تحليل: من شاشة التنفيذ نرى انه يعد 1 (ثم 12) (ثم 123) ويستمر نرى أن تكوين رسم بهذا الشكل يستحيل دون استخدام عبارات التكرار المتداخلة. فما نحتاجه هنا هو عبارة تكرارية خارجية تعد من 2 إلى 6 (أي تعد خمس عدات بقدر عدد الأسطر تعد) 2,3,4,5,6 (و عبارة تكرارية داخلية تبدأ بالعد من الواحد و شرط توقفها اقل من قيمة العداد الخارجي لكل عدة مثلا في العدة الأولى تكون قيمة المتغير $i=2$ في العداد الخارجي سيعد العداد الداخلي من واحد إلى اقل من 2 (أي سيعد 1) وفي العدة الثانية تكون قيمة المتغير $i=3$ في العداد الخارجي سيعد العداد الداخلي من واحد إلى اقل من 3 (أي سيعد 12) ويستمر إلى نهاية.

توضيح الخطوات :

1. خطوة رقم 1 تم تعريف متغيرين للعدادات. وفي خطوة رقم 2 عداد خارجي يعد من 2 إلى 6 وتتبعه الخطوات البرمجية رقم 3 و 4 و 5 لأنها داخل قوسيه ففي كل عدة له تحدث هذه الخطوات
 2. خطوة رقم 3 هي عبارة تكرارية داخلية تبدأ بالعد من الواحد إلى اقل من قيمة العداد الخارجي أي إذا عد العداد الخارجي 2 تعد هذه العبارة التكرارية 1 وإذا عد العداد الخارجي 3 تعد هذه العبارة التكرارية 12 وإذا عد العداد الخارجي 4 تعد هذه العبارة التكرارية 123 وإذا عد العداد الخارجي 5 تعد هذه العبارة التكرارية 1234 ويستمر
 3. خطوة رقم 4 تابعة للخطوة رقم 3 حتى تطبع قيم العداد وخطوة رقم 5 تابعة للخطوة رقم 2 حتى بعد كل طباعة ينزل المؤشر إلى سطر جديد لكي يطبع العبارة الجديدة على سطر آخر. جرب احذفها وشاهد ماذا سيحدث
- ❖ مثال: لو أردنا عداد يعد من 0 إلى 4 أي يطبع على شاشة التنفيذ من 0 إلى 4

```
#include<iostream>
Using namespace std;
main()
{
int i=0;
while( i<5)
{
cout<<i<<"\t";
i++;
} }
```

توضيح الخطوات :

1. خطوة رقم 1 عرفنا متغير `i` من نوع `integer` فأعطيناه القيم البدائية أربعة
2. خطوة رقم 2 هي عبارة تكرارية بما إننا نريد العد من الأربعة فأعطينا القيم البدائية أربعة. والعد يصل إلى الصفر معناه شرط التوقف أن يكون أكبر أو يساوي صفر `while(i>=0)` بما إننا نريد أن يعد خمس عدات بين صفر والأربعة لذلك يجب أن يكون مقدار نقصان واحد ليعد 0,1,2,3,4. لأن لو جعلنا مقدار النقصان 2 مثلا فسيعد العداد ثلاث مرات مرة 0,2,4 لذلك يجب التركيز في هذه المواضيع جيدا
3. خطوة رقم 3 طباعة قيمة عداد. وخطوة رقم 4 عداد النقصان يتناقص بمقدار واحد عند كل عدة

❖ مثال: لو أردنا عداد يعد من 0 إلى 4 أي يطبع على شاشة التنفيذ من 0 إلى 4

```
#include<iostream>
Using namespace std;
main()
{
int i=0;
do {
cout<<i<<"\t";
i++;}
while( i<5);
}
```

توضيح الخطوات :

1. خطوة رقم 1 عرفنا متغير `i` من نوع `integer` بما إننا نريد العد من الصفر فأعطينا القيم البدائية صفر
2. خطوة رقم 2 تمتد إلى خطوة رقم 5 هي عبارة تكرارية بما أن العد يصل إلى 4 معناه شرط التوقف أن يكون أقل من خمسة بما إننا نريد أن يعد خمس عدات بين صفر والأربعة لذلك يجب أن يكون مقدار الزيادة واحد ليعد 0,1,2,3,4. لأن لو جعلنا مقدار الزيادة 2 مثلا فسيعد العداد ثلاث مرات مرة 0,2,4 لذلك يجب التركيز في هذه المواضيع جيدا.



1. اكتب برنامج تكون مخرجاته ك التالي؟
2. اكتب برنامج اكتب برنامج لتقريب أي عدد يدخله المستخدم؟
3. اكتب برنامج عندما تضغط الي ارقام الشهور يطبع اسم الشهر مثلا 1 = January ؟
4. اكتب برنامج يطلب من المستخدم ادخال رقم ومن ثم يحدد هل الرقم زوجي ام فردي؟
5. اكتب برنامج لجمع الاعداد الفردية بين (0—100)