الكتاب التعليمي للبرمجة المهيكلة باستخدام لغة c++

تــــــــــــــــأليـــف

الطالبة / نور الهدى لطيف جاسم

الطالبة / هنــــد قاســــــم مثــــــنــــــــــــــــــــى

التدقيق العلمي

الدكتور/ نصير علي حسين القريشي

جامعة واسط /كلية التربية/قسم علوم الحاسبات

**الإهداء**

**إلى كل من أضاء بعلمه عقل غيره**

**أو هدى بالجواب الصحيح حيرة سائليه**

**فأظهر بسماحته تواضع العلماء**

**وبرحابته سماحة العارفين**

**والى كل من شجعني وآزرني بالفعل وبالكلمة (أمي وأبي ,أساتذتي ,زملائي وزميلاتي )**

مقدمة الكتاب

**اللهم لا علم لنا إلا ما علمتنا .**

**بسم الله والصلاة والسلام على رسول الله وعلى اله وصحبه وسلم والحمد لله رب العالمين .**

**يتناول هذا الكتاب لغة c++ بأسلوب مبسط جداً بالإضافة إلى شرح تفصيلي للبرامج الموجودة بحيث تسهل على القارئ يفهم ويتعلم ماهية السؤال والكيفية التي تمت كتابة البرنامج بها من حيث تسلسل الخطوات وغيرها , ولا بد من الإشارة إلى إن هذا الكتاب لا يحتوي على كل ما هو موجود بلغة c++ وإنما يحتوي على المواضيع التي تعتبر أساس للبداية مع البرمجة . حيث انه تم شرح المصفوفات وبرامج المصفوفات بطريقة سهلة الفهم على الطالب شرحت بشكل مخططات توضح عمل التكرارات وتسلسل الخطوات .**

**ومن الله التوفيق**

**مقدمة عن البرمجة و لغات البرمجة .**

**من الممكن تعريف البرمجة بأنها عملية كتابة تعليمات و أوامر لجهاز الحاسوب , لتوجيهه وإعلامه بكيفية التعامل مع البيانات أو كيفية تنفيذ سلسلة من الإعمال المطلوبة . تتبع عملية البرمجة قواعد خاصة باللغة التي اختارها المبرمج وكل لغة لها خصائصها التي تميزها عن الأخرى وتجعلها مناسبة بدرجات متفاوتة لكل نوع من أنواع البرامج والمهمة المطلوبة من هذا البرنامج .**

**كما إن للغات البرمجة أيضا خصائص مشتركة بحكم إن كل هذه اللغات صممت للتعامل مع الحاسوب .**

**ما المقصود بلغة البرمجة ؟**

**لغة البرمجة هي عبارة عن مجموعة من الأوامر تكتب وفق مجموعة من القواعد تحدد بواسطة لغة البرمجة ومن ثم تمر هذه الأوامر بعدة مراحل إلى إن تنفذ على جهاز الحاسوب .**

**تقسم لغات البرمجة إلى عدة مستويات على حسب صعوبتها وقربها من لغة الحاسب .**

* **لغة الآلة : وتسمى اللغة الثنائية حيث تتكون من سلسلة من 1,0 وهي اللغة الوحيدة التي يفهمها الحاسب الآلي , حيث تحول جميع اللغات إلى لغة الآلة حتى تتمكن معدات الحاسب الآلي من التفاهم معها ولأنها تتكون من 0,1 لذا فقد تميزت هذه اللغة بالصعوبة نظراً لما تتطلبه من حفظ ورقة في كتابة سلسلة طويلة من 0,1 بترتيب معين مما ينتج عنه أخطاء كثيرة من الترميز.**
* **لغة التجميع : وتسمى لغة الترميز حيث تستخدم الرموز للتعبير عن لغة الآلة , وذلك لمواجهة صعوبة الترميز ,ولغة الترميز أو التجميع لغة قريبة من لغة الآلة التي يفهمها الحاسب الآلي وتسمى هذه اللغات بلغات المستوى البسيط .**

**حيث يتم استعمال مختصرات و رموز يسهل حفظها وكتابتها لكل تعليمة من تعليمات لغة الآلة , حيث يقوم البرنامج المسمى Assembler بترجمة البرنامج المكتوب بلغة التجميع إلى لغة الآلة .**

* **اللغات العليا : وهي ( language, C language, c++language , SQL, HTML, Java Basic language, Pascal) سميت بهذا الاسم لأنه أصبح بإمكان المبرمج كتابة البرنامج دون معرفة تفاصيل كيفية قيام الحاسب بهذه العمليات , كمواقع التخزين وتفاصيل الجهاز الدقيقة , وتعبيرات لغات المستوى العالي إلى درجة كبيرة باللغة الطبيعية التي يستخدمها الإنسان في حياته للتواصل والتخاطب مع الآخرين .**

**ومن مميزات اللفات العليا التي تميزها عن لغات المستوى البسيط بالإضافة إلى ما سبق إن هذه اللغات غير مرتبطة بجهاز معين أي يمكننا تنفيذ البرنامج المكتوب بلغة من لغات المستوى العالي على أكثر من جهاز كما يمكن استخدام أكثر من لغة برمجية على جهاز معين كذلك فإن اكتشاف الأخطاء وتصحيحها أصبح أكثر سهولة بسبب سهولة قراءة البرامج وتتبعها وفهمها .**

**البداية مع البرمجة.**

**عند كتابة أي برنامج علينا أن نقوم بخطوات محددة تبدأ بتحليل المشكلة (البرنامج) ثم وضع خطوات لحل هذه المشكلة عن طريق استخدام الخوارزميات ثم تحول هذه الخوارزميات إلى مخططات ثم إلى شفرة (اللغة البرمجية التي تستخدم لحل المشكلة).**

**الخطوات المتبعة لحل البرنامج .**

* **الخوارزميات Algorithm .**
* **المخططات Flow chart .**
* **الشفرة Code .**

**Note : الخوارزميات والمخططات متشابهة في جميع لغات البرمجة , أما الشفرة تختلف من لغة إلى أخرى .**

* **الخوارزميات :- هي خطوات متسلسلة لحل مشكلة معينة و تتمثل هذه المشكلة في العلاقة بين المدخلات والمخرجات أي توضح العلاقة بين المدخلات للوصول إلى النتائج المتمثلة بالمخرجات .**

**تستخدم الخوارزميات لتبسيط وتسهيل الحل واختصار الوقت وليس الحصول على النواتج فمثلاً لدينا المعادلة y=x2\2x**

**نبدأ أولاً بحساب المقام ونتأكد أنه لا يساوي صفر (وهذا هو المقصود بتحليل المشكلة) ثم بعد ذلك نقوم بإيجاد ضعف x ومن**

**ثم حساب مربع x وأخيراً نقوم بحساب وبهذا نكون قد حصلنا ع قيمة y , وبهذا فأن الخوارزميات تستخدم لتبسيط الحل كما**

**في المعادلة السابقة فأننا لم نحصل على الناتج لكن حصلنا على الخطوات التي تمكننا من الحصول على الناتج** .

**الخطوات المتبعة لحل أي خوارزمية.**

**1 \_ لكل خوارزمية بداية و نهاية (Start & Stop or End ).**

**2\_ الإعلان عن المتغيرات والإعلان يكون للمدخلات والمخرجات(InPut & OutPut).**

**3 \_ قراءة المتغيرات أي أعطاء قيم للمدخلات فقط(InPut).**

**4 \_أجراء عملية المعالجة (مطلوب السؤال).**

**5\_ طباعة النواتج المتمثلة بالمخرجات (OutPut).**

**Note:\_ عند كتابة أي خوارزمية يجب الأخذ بنظر الاعتبار عدد المتغيرات الموجودة في السؤال (InPut & OutPut ) ونقصد بهذا هو تحليل المشكلة أو السؤال .**

**بعض الأمثلة على الخوارزميات.**

**Example 1 : Write algorithm to collect tow number .**

**Note:\_ نرمز للعدد الأول A والعدد الثاني B وناتج الجمع C أي تكون المعادلة C=A+B .**

Solution:

*Step 1 : Start.*

*Step 2 : Declare A,B,C .*

*Step 3 : Read A,B.*

*Step 4 : C=A+B.*

*Step 5 : Print C.*

*Step 6 : Stop.*

**Example 2 : Write algorithm to find average of 3 numbers** **.**

Solution:

*Step 1 :Start .*

*Step 3: Read A , B , C .*

*Step 4 : sum=A+B+C .*

*Step 5 : avg=sum / 3 .*

*Step 6 : Print avg .*

*Step 7 : Stop .*

Note**:\_ بما انه المطلوب أيجاد المعدل لـ ثلاث أعداد فأن عدد المتغيرات في هذا السؤال تكون خمس متغيرات A&B&C)) هي الإعداد الثلاثة , وبما انه يريد أيجاد المعدل فأننا نحتاج إلى متغير sum نضع فيه ناتج الجمع لهذه الإعداد ومتغير أخر avg للمعدل وذلك بتقسيم ناتج الجمع على عدد الإعداد** .

**Example 3 : Write algorithm to find the largest number of the tow numbers .**

Solution:

*Step 1 :**Start .*

*Step 2 : Declare**A , B , Max*

*Step 3 : Read A, B ,Max .*

*Step 4 : if (A>B)*

*Max =A .*

*else*

*Max =B .*

*Step 5 : Print Max .*

*Step 6 : Stop .*

Note**:\_ في هذا السؤال توجد احتمالية إما العدد الأول هو الأكبر أو بالعكس , لذا يجب أ، نكون جملة شرطية باستخدام if نختبر أي العددين اكبر فأنه إذا تحقق شرط if هذا يعني أن A هو العدد الأكبر وأن لم يتحقق فهذا يعني أن جملة else هي التي تتحقق وأن قيمة B هي الأكبر .**

**Example 4 : Write algorithm to order three numbers ascending order .**

**Note:\_ لنفترض أن الإعداد هي ( 2 نرمز له بالرمز A ) و (6 نرمز له بالرمز B) و (3 نرمز له بالرمز C ) , فأن ترتيب الإعداد سوف يكون ( 6 , 3 , 2) أي ( B , C, A ) لنلاحظ المخطط الآتي**

**هل الأول أصغر من الثاني ؟**

**نعم** **لا**

أصبح الأول هو الأصغر الأول ليس الرقم الأصغر

وفي مكانه الصحيح ولا نعرف مكانه الصحيح

**هل الثاني أصغر من الأول ؟ هل الثاني أصغر من** الثالث ؟

**نعم** **لا نعم** **لا**

الترتيب هو : الترتيب هو : أصبح الثاني هو الأصغر فأن الثالث هو الأصغر

الأول الأول في الموقع الأول في الموقع الأول

الثاني الثالث

الثالث الثاني نعود لمقارنة الأول مع الثالث و مقارنة الأول مع الثاني

**هل الأول أصغر من الثالث؟** **هل الثاني أصغر من الأول؟**

**نعم** **لا** **نعم** **لا**

الترتيب هو : الترتيب هو : الترتيب هو: الترتيب هو:

الثاني,الأول,الثالث الثاني,الثالث,الأول الثالث,الثاني,الأول الثاني,الأول,الثالث

Note**:\_ كل نعم تعني أن شرط if تحقق وكل لا تعني أني else هي التي تحققت .**

Solution:

Step 1 : Start .

*Step 2 : Declare A , B , C , num1 , num2 , num3 .*

*Step 3 : Read A , B , C .*

*Step 4 : if ( A < B && A < C ) .*

*num1=A*

*if ( B < C )*

*. num2=B*

*. num3=C*

*else*

*num2=C.*

*num3 =B .*

*else*

*if ( B < C )*

*. num1=B*

*else*

*num1=C*

*if ( A < C )*

*. num2=A*

*. num3=C*

*else*

*. num2=C num3=A .*

*if ( B < A )*

*. num2=B*

*. num3=A*

*else*

*num2=A.*

*num3 =B .*

*Stop 5 : Print num1 , num2 , num3 .*

*Step 6 : Stop .*

**Example 5 : Write algorithm to swap between X , Y.**

Note**:\_ لنفترض أن قيمة x هي x=5 و قيمة y=2 المطلوب هو أن تكون قيمة x=2 و قيمة y=5 .**

**للحصول على هذا التبديل نحتاج إلى متغير ثالث وليكن z هذا المتغير ننقل فيه قيمة احد المتغيرين فمثلاً ننقل قيمة x في z سوف تصبح x لا تحتوي على أي قيمة بإمكاننا أن ننقل قيمة y إلى x وبهذا قد نكون قد حققنا جزء من مطلوب السؤال ومن ثم ننقل قيمة z والتي هي في الأصل تعتبر قيمة x ننقلها إلى y وهكذا نكون قد بدلنا قيمتي . x , y**

**لاحظ المخطط الأتي:**

**2- نققل قيمة** X = Y

Y X

5 2

2 5

3**- ننقل قيمة** Y = Z 1**- ننقل قيمة** Z = X

Z

5

الصيغة الرياضية للتبديل هي :

Z = X // Z=5

X = Y // X= 2

Y = Z // Y=5

حسب الملاحظات فأن الخوارزمية هي :

*Step 1 : Start .*

*Step 2 : Declare X , Y , Z .*

*Step 3 : Read X , Y .*

*Step 4 : Z = X*

*X=Y*

*Y = Z*

*Step 5 : Print X ,Y .*

*Step 6 : Stop .*

**المخططات flow chart .**

**هي عبارة عن مجموعة من الأشكال الهندسية ترتبط مع بعضها عن طريق الأسهم التي توضح اتجاه سير الخوارزمية و أيضاً توضح عمل الخوارزمية (أي تحويل الخوارزمية إلى مخططات ) وتمتاز المخططات بالوضوح أكثر من الخوارزمية خاصة في الجمل الشرطية والتكرارات .**

**الأشكال المستخدمة في عمل المخططات.**

**الشكل البيضوي يستخدم هذا الشكل في بداية المخطط (Start) وفي نهاية المخطط (Stop) .**

**المستطيل يستخدم هذا الشكل للإعلان عن المتغيرات وأيضاً يستخدم في جزء المعالجة إذا كانت العملية المطلوبة**

**عملية حسابية أو أي عملية ما عدا الجمل الشرطية .**

**الشكل ألمعيني يستخدم هذا الشكل في الجمل الشرطية حيث يكون هناك احتمالين للناتج .**

**متوازي الإضلاع يستخدم هذا الشكل لإدخال القيم أو قراءة القيم وأيضاً لطباعة النواتج أو المخرجات .**

**الأسهم تستخدم للتوصيل بين الإشكال .**

**Example 1 : Drew the flow chart to collect tow number .**

Note**:\_** **خوارزمية هذا السؤال سبق وأن تطرقنا لها عند شرح الخوارزميات , أما المخطط فهو كالأتي**

**البداية**

**الإعلان عن المتغيرات**

Declare A , B, C

**قراءة المتغيرات**

Read A , B

**جزء المعالجة**

C = A + B

**طباعة الناتج**

Print C

**النهاية**

**Example 2 : Draw the flow chart to find average of 3 numbers .**

**البداية**

**الإعلان عن المتغيرات**

Declare A,B,C,sum,avg A,B,C,SUM,

Read A,B,C

**قراءة المتغيرات**

Sum=A+B+C

Avg=sum/3

**جزء المعالجة**

**طباعة النواتج**

Print avg

**النهاية**

**Example 3 : Draw the flow chart to find the largest number of the tow numbers .**

Max = A

Print Max

Max = B

(A>B)

Read A, B

Declare A,B,max

**إذا تحقق الشرط فأن الأكبر T F إذا لم يتحقق الشرط فأن هو A الأكبر هو B**

**. Example 4 : Draw the flow chart to swap between X , Y**

Declare X,Y,Z

Read X, Y

Z=X

X=Y

Y=Z

Print X, Y

**مدخل إلى البرمجة بلغة c++.**

* **الرموز المستخدمة في لغة c++ .**
* **الحروف الانكليزية الصغيرة والكبيرة .(A, B, a ,c) .**
* **الأرقام (1,2,3,…..) .**
* **رموز خاصة وهي .**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-** | **+** | **\*** | **\** | **&&** | **||** | **[]** | **()** | **{}** |
| **<>** | **>** | **<** | **<<** | **>>** | **!** | **/** | **=** | **\_** |
| **#** | **%** | **,** | **;** | **"** | **'** | **&** | **=!** | **:** |

**-:Note أن لكل رمز من هذه الرموز وظيفة خاصة يؤديها في حال استخدامها في البرنامج .**

* **الكلمات المستخدمة في لغة c++ .**

**هناك نوعين من الكلمات التي تستخدمها في لغة c++ .**

**أولا: الكلمات المحجوزة في لغة c++ .**

**تستخدم هذه الكلمات التي تكون معروفة بالنسبة لمترجم لغة c++ لأداء وظيفة معينة تماما كالرموز فأن لكل كلمة وظيفة تؤديها ولا يجوز استخدامها في غير الوظيفة التي تقوم بها مثلا كاستخدامها لأسم متغير أو اسم دالة وغيرها .ومن هذه الكلمات هي .**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Long** | **Double** | **String** | **Char** | **Float** | **Int** |
| **Return** | **Switch** | **Else** | **If** | **While** | **Do** |
| **Private** | **Protected** | **Const** | **For** | **Continue** | **Default** |
| **Friend** | **Struct** | **Main** | **Void** | **Class** | **Public** |
| **Static** | **Typedef** | **Union** | **List** | **Sizeof** | **Break** |
| **Volatile** | **Catch** | **Inline** | **Delete** | **Goto** | **Auto** |
| **Entry** | **Strcpy** | **Strlen** | **Signed** | **Register** | **Pascal** |

**ثانيا:الأسماء أو الكلمات التعريفية (المتغيرات ).**

**تستخدم الأسماء التعريفية لنشير بها إلى شيء معين , ويطلق على هذه الأسماء بالمتغيرات . هناك بعض القيود أو الشروط**

**الواجب مراعاتها عند اختيار اسم المتغير وهي .**

* **أن يكون اسم المتغير مكون من مجموعة من حرف واحد أو مجموعة من الأحرف بشكل متصل يبدأ بحرف أو الخط التحتي (\_).**
* **أن لا يحتوي اسم المتغير على الرموز عدا الخط التحتي (\_).**
* **أن لا يكون الاسم احد الكلمات المحجوزة .**
* **أن يكون الاسم مكتوب باللغة الانكليزية حصراً.**

1. **X\_AB**
2. **A**
3. **Name1**

**بعض الأمثلة على كتابة المتغيرات بالصورة الخاطئة .**

1. **1abc / لا يجوز أن تبدأ برقم .**
2. **A! / لا يجوز أن يحتوي على رموز .**
3. **$XYZ / لا يجوز استعمال حروف غير الانكليزية**

**Note:- أن لغة c++ تفرق بين الحروف الكبيرة و الصغيرة فمثلاً المتغير abc يختلف عن المتغير ABC وأيضاً يختلف عن المتغير Abc فأن المترجم الخاص بلغة c++ يعامل كل متغير من هذه المتغيرات بشكل مستقل عن الأخر .**

**أنواع المتغيرات .**

**كل متغير يستخدم في البرنامج في لغة c++ يجب أن يصنف ضمن نوع بيانات خاص , وسنتطرق لبعض هذه الأنواع من البيانات .**

* **Integer تستخدم للمتغيرات التي تحمل قيم عددية صحيحة وعند استخدامها تكتب بشكل مختصر int .**
* **Float تستخدم للمتغيرات التي تحمل قيم عددية عشرية وعند استخدامها تكتب كما هي .**
* **Char تستخدم للمتغيرات التي تحمل حرف واحد (single character) وعند استخدامها تكتب كما هي .**
* **String تستخدم للمتغيرات التي تحمل سلسلة من الأحرف وعند استخدامها تكتب كما هي .**
* **Double تستخدم للمتغيرات التي تحمل قيم عددية صحيحة كبيرة وعند استخدامها تكتب كما هي .**

**Note:- سيتم شرح string & char بشكل تفصيلي في الفصول المتقدمة .**

**تعريف المتغيرات .**

* **تعرف المتغيرات التي تحمل قيم عددية صحيحة .**

**Int x=9 ;**

* **تعريف المتغيرات التي تحمل قيم عددية عشرية .**

**Float y=2.5 ;**

* **تعريف المتغير الذي يحمل حرف واحد .**

**Char z='A' ;**

* **تعريف المتغير الذي يحمل سلسلة من الأحرف.**

**string k = "Ali " ;**

* **العمليات الحسابية في لغة c++ .**
* **- للطرح أو الإشارة السالبة .**
* **+ للجمع .**
* **\* للضرب .**
* **/ للقسمة .**
* **% باقي القسمة .**
* **-- نقصان .**
* **++ زيادة .**
* **العمليات العلاقية في لغة c++ .**
* **> أكبر من .**
* **< أصغر من .**
* **>= أكبر أو يساوي .**
* **<= أصغر أو يساوي .**
* **== يساوي .**
* **!= لا يساوي .**
* **العمليات المنطقية في لغة c++ .**
* **&& and )).**
* **|| (or) .**
* **! (not).**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **! Y** | **! X** | **X || Y** | **X && Y** | **Y** | **X** |
| **1** | **0** | **1** | **0** | **0** | **1** |
| **0** | **1** | **1** | **0** | **1** | **0** |
|  |  | **1** | **1** | **1** | **1** |
|  |  | **0** | **0** | **0** | **0** |

**من الجدول أعلاه نلاحظ أن 1 ) ) تعني T)) أي أن الشرط متحقق وأن (0) تعني (F) أي أن الشرط غير متحقق .**

**في جملة && حتى تحقق يشترط أن تكون قيمX,Y هي (1) أو (T) , أما جملة || حتى تتحقق يكفي أن تكون قيمة أحد المتغيرين (1) أو (T) أي إما قيمة X تساوي (1) أو قيمة Y تساوي (1) , أمل بالنسبة للنفي ! فأنه عكس القيم أي أذا كانت قيمة X تساوي (1) فأن نفي X هو (0) وكذلك بالنسبة Y .**

**وظيفة بعض الرموز المستخدمة في C++ .**

**b\← تعني الرجوع بمقدار خانة واحدة.**

***\n← تعني سطر جديد.***

**t\← ترتيب أفقي.**

**o\← للقيمة الخالية.**

**r\← تعني علامة الرجوع.**

**v\← ترتيب عمودي.**

**e\← تقديم صفحة.**

**a\← إصدار صوت عند التنفيذ.**

**\_الكيفية التي يتم فيها كتابة برنامج بلغة C++.**

**لكتابة إي برنامج بلغة C++ نتبع الهيكلية التالية:**

**#include<iostream.h>**

**Void main ()**

**{**

**// definition of variables**

**// read and print variable**

**}**

**#include :\_ عبارة عن موجه للمترجم لكي يقوم بإخراج ملف رئيسي ضمن الملف الأساسي ويجب أن توضع بين<>.**

**iostream.h :\_io مختصر لinput/out put إي الإدخال والإخراج .**

**Stream :\_ مكتبة قياسية خاصة في الإدخال والإخراج .**

**\_ : hإي header file .**

**: #include<iostream.h> \_ تسمى منطقة التعريفات العامة واستدعاء مكاتب للبرامج حيث يتم كتابة جميع المكاتب التي سنحتاج إليها واقل البرامج وكذلك المتغيرات التي تعرف بشكل عام لكل البرامج والسجلات والدوال مثلا (cout , cin)في لغة C++ تقع ضمن مكتبة <iostream.h>.**

**Void main() :\_ (void) تعني إن هذه الدالة لا ترجع إي قيمة .**

**Main():.تعتبر الدوال من أهم مقومات البرنامج في لغة C++ ونجد إن البرنامج يمكن إن يتألف من دالة واحدة أو أكثر وانه لابد لكل دالة من اسم يدل عليها عند استخراجها واستدعائها فأن.Main()هي الأساس لأي برنامج ولن يعمل البرنامج إلا بدونها وهي بمعنى دالة مستقلة بنقل نظام التشغيل التحكم إليها وهي الجزء الأساسي في برنامج C++ فأن المترجم يبحث عن هذه الدالة أولا لتنفيذها , إما إذا لم تكن موجودة فتظهر رسالة خطأ توضح ذلك .**

**(): مهمة جدا فهي تعني إن هذا الاسم عن دالة وليس متغير.**

**{}:.هما جسم الدالة الرئيسية وتكتب بداخله وتكتب بداخله تعريف للمتغيرات المستخدمة في البرنامج و الاكواد وأيضا قراءة المتغيرات وطباعتها تتم كتابتهم داخل جسم الدالة .**

**هناك ثلاث إشكال للدالة الرئيسية:.**

**1**

**Main()**

**{**

**............**

**............**

**}**

**2**

**Int maim()**

**{**

**...........**

**...........**

**Return 0;**

**}**

**3**

**Void main()**

**{**

**..............**

**..............**

**}**

**وعند استخدام دالة int main() فيجب استخدام return 0 في البرنامج وذلك من اجل إرجاع قيمة عددية صحيحة إي إن int تستخدم لإرجاع قيمة عددية صحيحة إي إن int تستخدم لإرجاع قيمة عن طريق return .**

**إما عند استخدام الـvoid main() فهذا يعني إن البرنامج لا يرجع إي قيمة .**

**Cout :\_ هي اختصار لـcourse output)) وهي تعتبر دالة أو كائن يقوم بعرض ما يأتي بعده على شاشة التنفيذ**

**>> يسمى معامل الإخراج يستخدم مع الـ cout لعرض النتائج ويكتب بالصيغة الآتيةcout<<.**

**\_ : Cin هي اختصار لـcourse input)) وهي تعتبر دالة أو كائن يستخدم لإدخال قيم المتغيرات المعرفة في البرنامج .**

**>> معامل الإدخال يستخدم لإدخال قيم المتغيرات ويكتب بالصيغة cin>>.**

**;)): \_ الفارزة المنقوطة يجب إن تكتب عند نهاية كل جملة تتطلب التنفيذ فهي تعني إن الأمر انتهى وعليه إن يذهب للأمر التالي .**

**ملاحظات مهمة**

**1-لغة C++ تفرق بين الحروف الكبيرة والصغيرة عند كتابة البرنامج فأن الأوامر والدوال والأسماء يجب إن تكتب بالحروف الصغيرة حصرا.**

**2- هناك كلمات محجوزة في لغةC++ ولا تستخدم هذه الكلمات إلا للوظيفة المخصصة لها ولا يجوز استخدامها**

**في غير وضيفتها.**

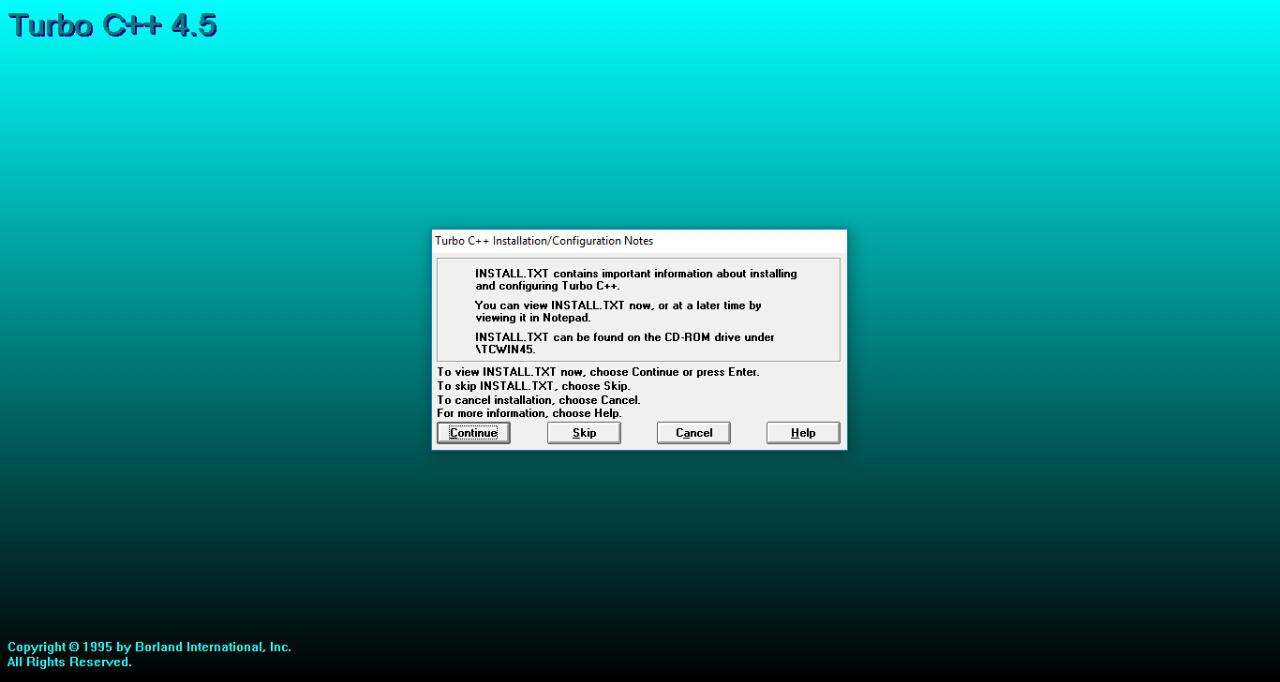
**3- إن لغةC++ لا تفرق بين وجود المسافات أو عدم وجودها عند كتابة البرنامج إلا في تعريف اسم المتغيرات.**

**4- كل جملة تكتب بين علامتي اقتباس تظهر الجملة كما هي في شاشة التنفيذ من دون إي تغيير.**

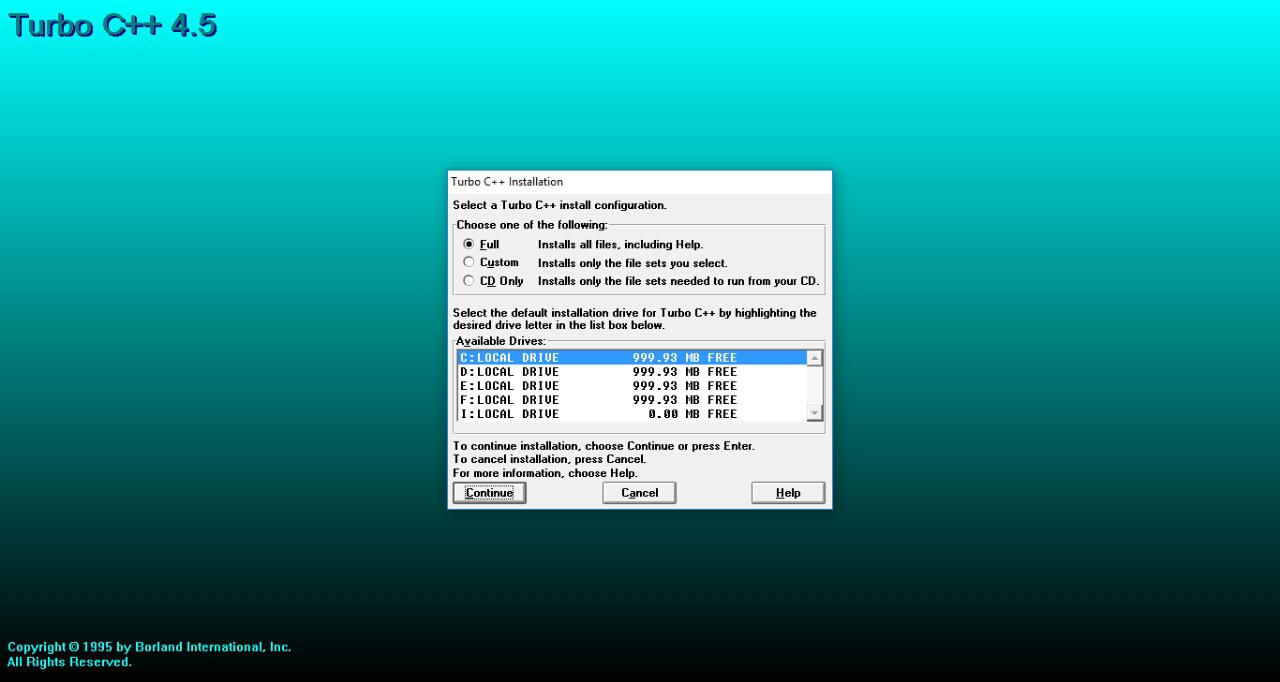
**5- // كل شيء يكتب بعد الشرطة المائلة يعتبر توضيح للسؤال وتسمى التعليقات وتساعد القارئ على فهم ماذا تعنيه هذه الخطوة أو السؤال ويمكن إن تكتب في إي مكان أو السؤال ويمكن إن تكتب في إي مكان في البرنامج ولا يكون لها اثر على البرنامج فهي لا تعتبر جزء من هيكلية البرنامج فأن الـcompiler يتجاهلها إثناء تنفيذ البرنامج ويتجاهل السطر الذي توجد فيه أيضا . وأيضا هناك نوع أخر من التعليقات التي تعلق تنفيذ جزء من الجملة وهي /\* ويجب أن توضع في بداية الجملة المراد إيقاف تنفيذها وفي نهايتها .**

**خطوات تنصيب برنامج C++ .**

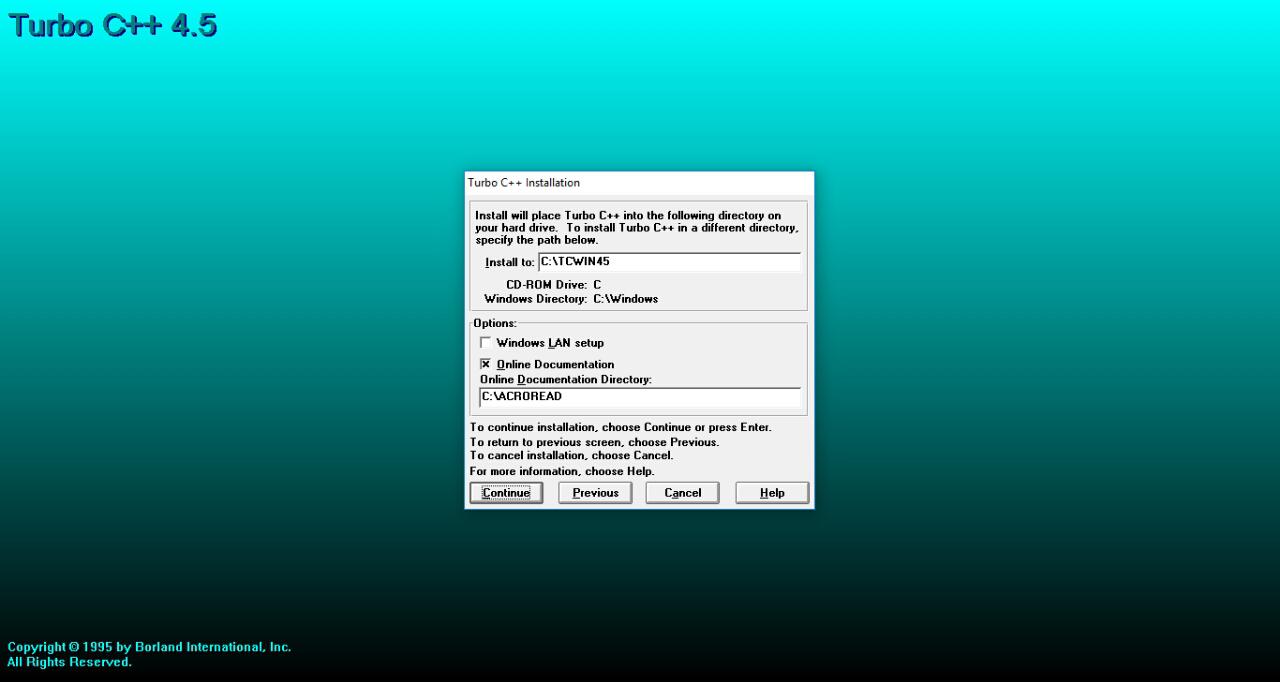
**الخطوة الأولى :**

****

**الخطوة الثانية :**

****

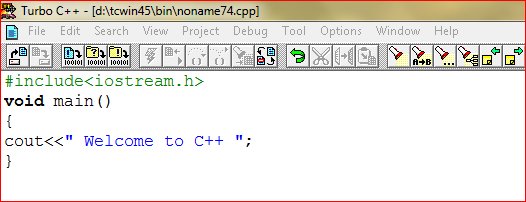
**الخطوة الثالثة :**

****

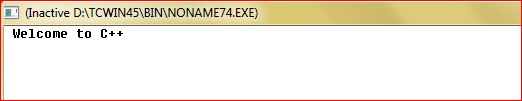
تحديد مكان التنصيب

**Q1 : Write a program to print the statement Welcome to C++ .**

**Solution :**

****

**Execution screen .**

****

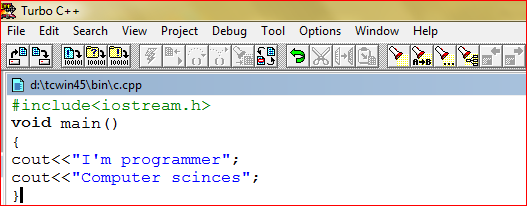
**البرنامج واضح جداً فهو يظهر على شاشة التنفيذ عبارة Welcome to C++ ولهذا استخدمنا cout<< لإظهار النتائج فقط .**

**Q2: Write a program to print the statement**

**I'm programmer**

**Computer sciences**

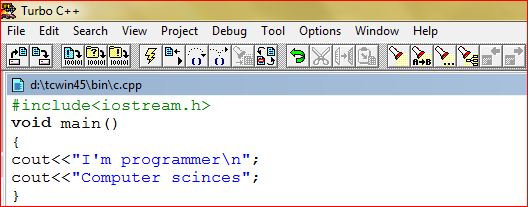
**Solution :**

****

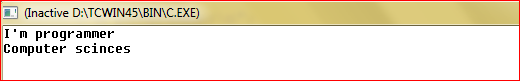
**Execution screen .**

**\_ :Note إن البرنامج يقوم بطباعة الجمل في سطر واحد ولأجل تلافي ذلك وطباعة كل جملة في سطر هناك طريقتين: **

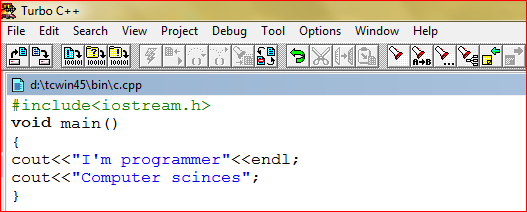
**الطريقة الأولى : . هي استخدام "\n"عند نهاية الجملة الأولى حتى يقوم بتنفيذ الجملة الأولى ثم ينفذ الجملة الثانية في سطر جديد.**

****

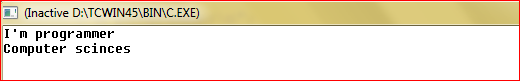
**Execution screen .**

****

**الطريقة الثانية : . هي استخدام الendl; هي تعني نهاية السطر الذي توجد فيه إي عند تنفيذ السطر الذي ينتهي بendl; ينزل إلى سطر جديد وينفذ الجملة الثانية.**

****

**Execution screen .**

****

**Q3: Write a program to collect tow numbers .**

**Note : \_ لحل أي سؤال يحتوي على معادلة أو أي عملية رياضية أو منطقية أو أي فكرة فأنه يتطلب خطوات متسلسلة لحل أي سؤال أولها كتابة خوارزمية السؤال ومن ثم تحويلها إلى برنامج .**

**Solution :**

**بداية البرنامج Step 1 : start**

**إعلان عن المتغيرات Step 2 : Declear X , Y , Z**

**إدخال البيانات Step 3 : Read X , Y**

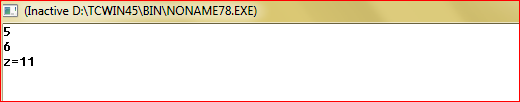
**معالجة البيانات Z=X+Y: Step 4**

**طباعة الناتج print Z Step 5 :**

**النهاية Step 6 : end**

****

**Execution screen .**

****

**شرح خطوات البرنامج**

**Int x,y,z; في هذه الخطوة عرفنا 3 متغيرات من نوع int إي إن القيمة التي ندخلها يجب إن تكون صحيحة .**

**ويجب إن تنتهي هذه الجملة بالفارزة المنقوطة باعتبار إن الجملة انتهت وعليه إن ينفذ الجملة التي بعدها وهي**

**على قيم لذا استخدمنا إيعاز cin لإدخال قيم x,y لأداء عملية الجمع وأيضا يجب إن تنتهي بالفارزة المنقوطة حتى يتم تنفيذ الجملة التي بعدها وهي.**

**Z=x+y;:. هي الخطوة المطلوبة في السؤال جمع قيمة x مع y ويضع الناتج في قيمة z وأيضا يجب إن تنتهي بالفارزة المنقوطة حتى ينفذ الجملة التي بعدها وهي.**

**Cout<<z; :. في هذه الخطوة يطبع قيمة z والتي تعتبر ناتج الجمع وكذلك يجب إن تنتهي بالفارزة المنقوطة.**

**Q4 : Write a program performs mathematical operation on variables .**

**الجمع addition**

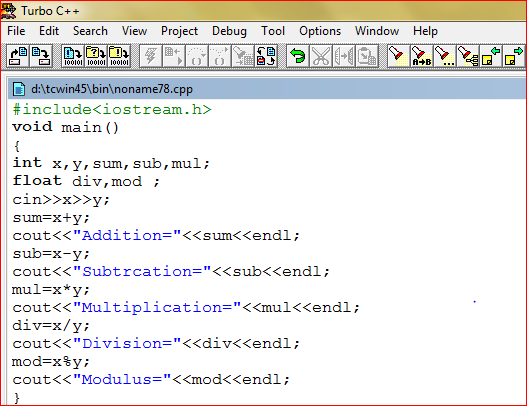
**الطرح subtraction**

**الضرب multiplication n**

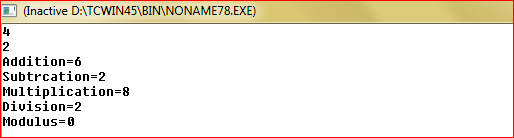
**القسمة division**

**باقي القسمة modulus**

**Solution :**

****

**Execution screen .**

****

* **ترتيب الأسبقيات في العمليات الرياضية والمنطقية :**

1. **()**
2. **^**
3. **\*mod ,and, / ,**
4. **+ or , - ,**
5. **=> , < ,**

**مثال :- A= B\*C % D +E/ F-G**

**الخطوة الأولى نحسب B\*C ثم ناتج الضرب نحسب الـ % مع ال D في الخطوة الثانية, إما الخطوة الثالثة نحسب E/f وناتج القسمة مع ناتج الخطوة الثانية نجمعه مع وناتج الخطوة الرابعة نطرحه من Gوبالتالي نحصل على قيمة A.**

* **استخدام بعض الصيغ الرياضية في C++**

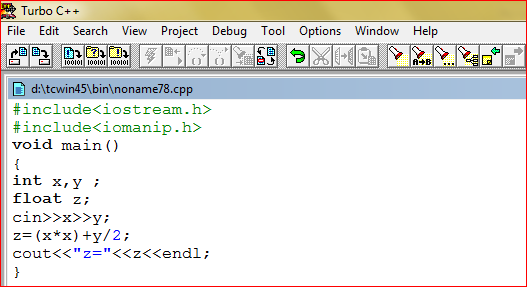
**عند حساب بعض المعادلات الرياضية في c++ وان كانت هذه المعادلات تحتوي على ex أو ln أو log أو وغيرها فيجب تحويلها إلى صيغة تكون مفهومة لدى complierفي c++ حسب الجدول التالي :**

|  |  |
| --- | --- |
| **C++ expression** | **Mathematical expression** |
| **Exp(n)** | **en** |
| **Log10(x)** | **Log(x)** |
| **Log(x)** | **Ln(x)** |
| **Sin(x)** | **Sin(x)** |
| **Pow(x,n)** | **xn** |
| **Sqrt(x)** | **x√** |

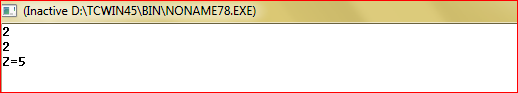
**علما إن هذه الصيغ لا تعمل إلا بوجود مكتبه math.h .**

**Q5 : Write a program that computes the following equation z=x2+y/2.**

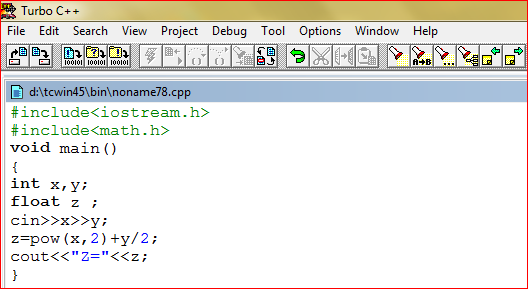
**Solution :**

****

**Execution screen .**

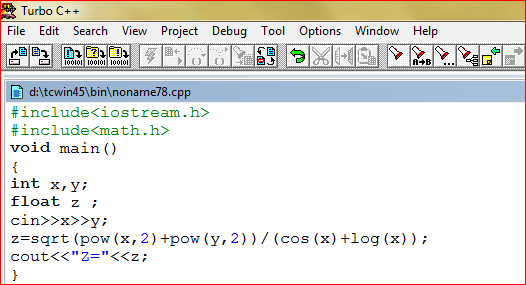
****

**هناك طريقة ثانيه لحساب هذه المعادلة :**

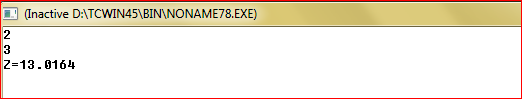
****

**Q6: Write a program that computes the following equation z**

**Solution :**

****

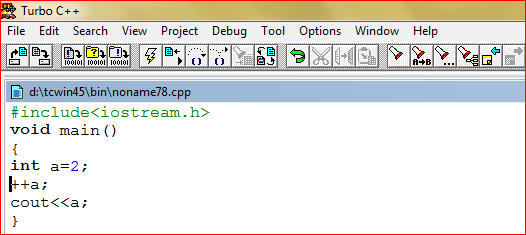
**Execution screen .**

****

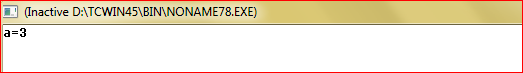
**Prefix & postfix .**

**Prefix صيغتها ++a او - -a**

**++aتعني زيادة قيم aبمقدار واحد أولا ثم ينفذ الخطوة إي لو كانت قيمة a=2 في البرنامج التالي :**

****

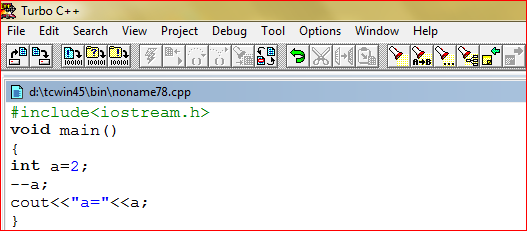
**Execution screen .**

****

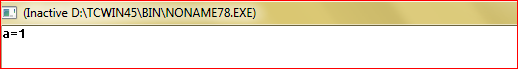
**إي إن مخرجات البرنامج التالي هي " 3 " إي قيمة a تغيرت وأصبحت " 3 "**

**أما ((--aتعني نقصان قيم aبمقدار واحد أولا ثم ينفذ الخطوة إي لو كانت قيمة a=2 في البرنامج التالي :**

**Execution screen .**

****

**Execution screen .**

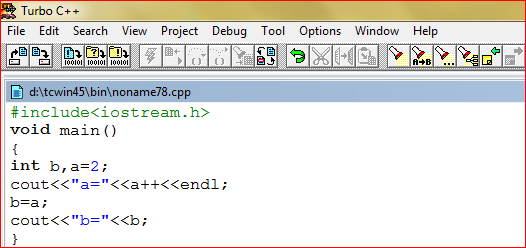
****

**إي إن مخرجات البرنامج التالي هي " 1 " إي قيمة a تغيرت وأصبحت " 1 "**

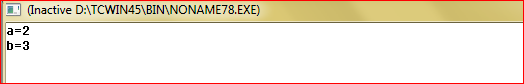
**postfix**

**صيغتها a++ او a--**

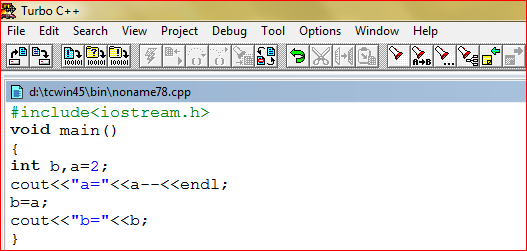
**a++ :- تعني تنفيذ الخطوة إي عرض قيمة aومن ثم زيادة قيمتها بمقدار 1 اي لو كانت قيمة a=2 حسب البرنامج التالي :**

****

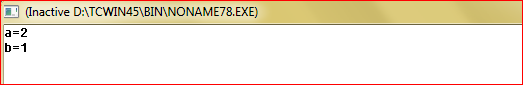
**حيث إن مخرجات البرنامج التالي هي 2 و 3 Execution screen .**

****

**إماa--) ) :- تعني تنفيذ الخطوة إي عرض قيمة aومن ثم نقصان قيمتها بمقدار 1 إي لو كانت قيمة a=2 حسب البرنامج التالي :**

****

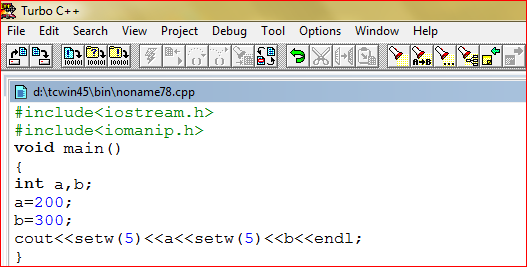
**حيث إن مخرجات البرنامج التالي هي 2 و1 Execution screen .**

****

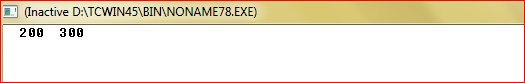
**Manipulator function متحكمات المخرج**

**هي مؤشرات توجه المترجم لاظهار المخرجات على هئية معنيه ومن اهمها :**

**أولا :- setw : هي اختصار لكلمة " setwidth " تقوم بتحديد حقول للكلمات المخرجة**

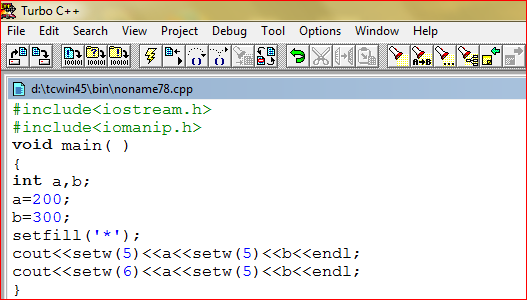
****

**Execution screen .**

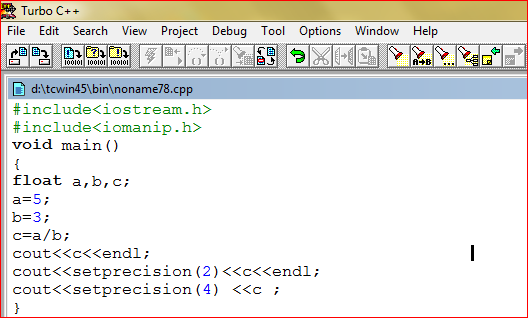
****

**اي يترك من ضمن مواقع يعرض قيمه a ومن ضمن مواقع ثانيه يعرض قيمة b**

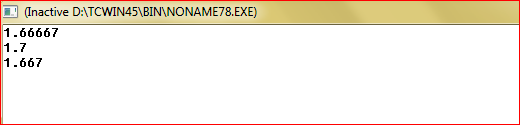
* **دالة setfill :-تستخدم هذه الدالة مع دالة setwلملئ الفراغات إن وجدت .**

****

**Setprecision :-تستخدم هذه الدالة لحساب عدد المراتب لأي عدد .**

****

**Execution screen .**

****

**الجمل الشرطية في c++**

**كثيرا ما نتخذ قراراتنا باستخدام الشرط فعندما نقول " إذا حصل هذا الشيء عندما سأفعل كذا وكذا " فانك تستخدم شرطا تبني على تحقيقه قرارا في تنفيذ مجموعة من الأشياء أو عدم تنفيذها . لنتعرف كيف تستطيع إن نفعل ذلك باستخدام c++**

**If ( expression or condition ;**

**وفي حالة وجود جمله أو تعليمه واحده if (expression or condition) //**

**statement**

**condition**

**Ex: if (x>y)// condition**

**Cout<<x;//statement;**

**الهيكلية التي تتم فيها كتابة if**

**If (expression or condition)**

**{**

**Statement 1;**

**Statement 2;**

**Statement 3;**

**.**

**.**

**Statement n;**

**}**

**Ex: if(x>y)**

**{**

**x+=y;**

**cout<<x;**

**}**

**حيث نلاحظ في هيكلية بناء ifفي الحالة الأولى والثانية انه تقوم جملة ifالشرطية بنقل تسلسل تنفيذ البرنامج إلى جملة أو سلسلة من الجمل كما في الحالة الثانية , عندما يتحقق الشرط تكون النتيجة True إي ينفذ الجملة أو مجموعة الجمل التي تكون داخل الشرط إما إذا لم يتحقق الشرط (شرط if) فانه من الطبيعي إن يتحقق شرط Else حيث إن لكل if يوجد لها Else والتي تعتبر الخيار الثاني لتحقيق الشرط وتنفيذ الجمل ,لاحظ البرنامج التالي**

**Ex: if(x<y)**

**Cout<<x;**

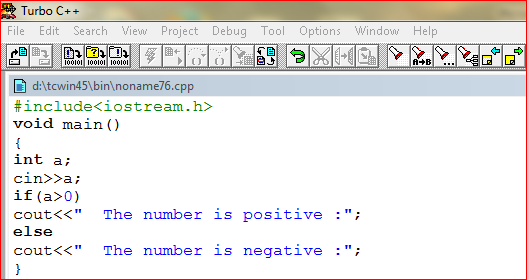
**else**

**cout<<y;**

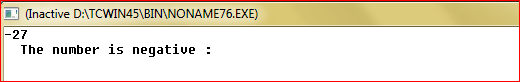
**نلاحظ في هذا الشرط انه مطلوب إيجاد اصغر عدد من بين عددين فإذا كان x هو الأصغر فإنه يقوم بتنفيذ جملة if وطباعة قيمة x إما إذا لم يكن x هو الأصغر فانه يهمل جملة if وينفذ جملة else وبهذا يكون قد حصل على العدد الأصغر وألان سوف نأخذ بعض الأمثلة للتوضيح أكثر**

**Q 1: Write a program enters the number entered by the user and test whether positive or negative**

**Solution :**

****

**Execution screen .**

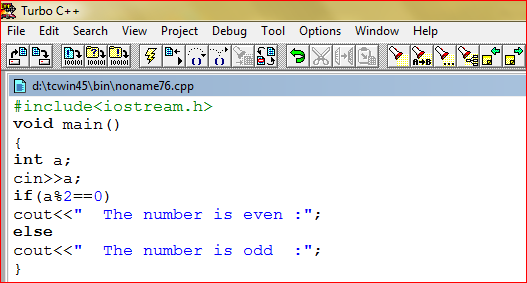
****

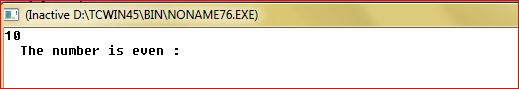
**في هذا السؤال لو أدخلنا قيمة a=2 فان شرط if في هذه الحالة يتحقق باعتبار 2 اكبر من 0 فانه سوف يطبع عبارة "a is positive"**

**إما إذا أدخلنا a=-1 في هذه الحالة نتيجة شرط if تكون false لان ((-1 ليس اكبر من 0 ولهذا فانه يطبع عبارة "a is negative"**

**Q2 : Write a program enters the number entered by the user and test whether even or odd .**

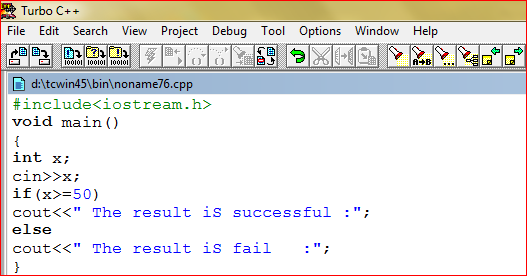
**Solution :**

** Execution screen .**

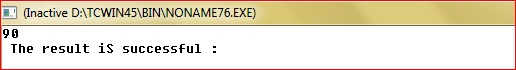
****

**Q 3 : Write a program enters the result and test whether successful or fail .**

**Solution :**

****

**Execution screen .**

****

**if & else if**

**if (condition 1)**

**statement 1;**

**if else (condition 2)**

**statement 2;**

**if else (condition 3)**

**statement 3;**

**else**

**statement 4;**

**حيث condition1 هو الشرط الأول و statement1 هي التعليمة التي يتم تنفيذها إذا تحقق الشرط الأول ويهمل بقية الشروط والجمل**

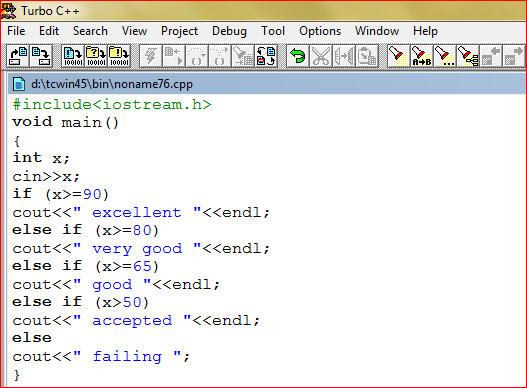
**إما condition2 هو الشرط الثاني و statement2 هي التعليمة التي يتم تنفيذها إذا لم يتحقق الشرط الأول وتحقق الشرط الثاني وأيضا يهمل بقية الشروط**

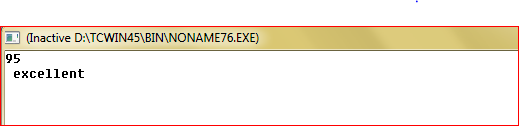
**إما condition3 هو الشرط الثالث و statement3 هي التعليمة التي يتم تنفيذها إذا تحقق الشرط ولم يتحقق الشرط الأول والثاني**

**إما else فإذا لم يتحقق إي شرط من الشروط فانه يجب إن يحقق تعليمة else وينفذ**

**Q 4 : Write a program enters the result and prints appreciation .**

**Solution :**

** Execution screen .**

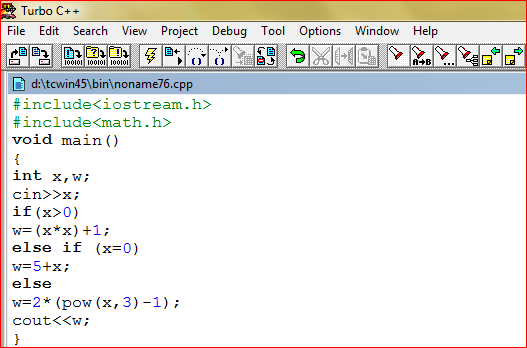
****

**Q5 : Write a program that computes the following equation .**

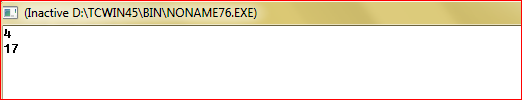
**x2+1 : x>0**

**w= x+5 : x=0 2x3-1 : x< 0**

**Solution :**

****

**Execution screen .**

****

**إذا كانت قيمة x المدخلة اكبر من 0 إي الحالة أو الشرط الأول فانه ينفذ التعليمة الأولى (x2+1)**

**إما إذا كانت قيمة x المدخلة هي 0 إي الشرط الثاني فانه ينفذ التعليمة الثانية (x+5)**

**إما إذا كانت قيمة x المدخلة اصغر من 0 وتعتبر هذه حالة else فانه ينفذ التعليمة الثالثة 2x3-1 .**

**التكرارات loops**

**عند كتابة البرنامج قد نحتاج إلى تكرار مجموعة من التعليمات ( عدد مرات معين ) لنأخذ المثال التالي للتوضيح أكثر , لو أردنا طباعة الأرقام من 1 إلى 100 فهل من الطبيعي إن نستخدم تعليمة cout مئة مرة ؟ لكي نعرض الأرقام من 1 إلى 100في شاشة التنفيذ , إن استخدام أسلوب التكرار يحل مثل هذه المشاكل وهنالك ثلاث إشكال من أساليب التكرار سنتعرف عليها تباعا وهي (for ,while ,do while)**

**إن من ميزات استخدام التكرار في البرنامج التي تتطلب تكرار مجموعة من الجمل أو التعليمات هي تقليل**

**خطوات البرنامج وأيضا توفير مساحة خزنيه في الذاكرة.**

**أسلوب التكرار باستعمال (for)**

**الصيغة العامة لاستخدام for**

**For( initialization; condition; update)**

**إن أسلوب التكرار باستخدام for يعتبر من أسهل أساليب التكرار وأكثر الحلقات التكرارية استخداما إذ تحتوي بداخلها على عدد يسمح بتكرار الجمل بعدد معين من التكرارات ولتوضيح هذه العبارة نتعرف على المصطلحات الموجودة في هيكلية for وماذا تعني كل واحدة**

**Initialization : القيمة الابتدائية التي يبدأ منها العد**

**Condition : شرط يقيد حركة for وغالبا ما يحتوي على قيمة نهائية , إي يعتبر شرط التوقف**

**Update : مقدار الزيادة أو النقصان في العداد حسب شرط السؤال**

**Statement : الجملة المراد تكرار تنفيذها حسب العداد**

**لنأخذ المثال التوضيحي التالي**

**for (i=0;i<3;i++)**

**حيث إن (i=0) إي بداية العد أو إن التكرار يبدأ من الصفر وان (i<3) تعني انه عندما تصل قيمة i إلى 3 يتوقف عن التكرار ويخرج من حلقة for وان (i++) تعني مقدار الزيادة بمقدار واحد**

**كما يمكن كتابة for بثلاث إشكال وهي**

**for(i<=10;) // هنا لم نعطي قيمة ابتدائية للعداد وكذلك لم نحدد مقدار الزيادة**

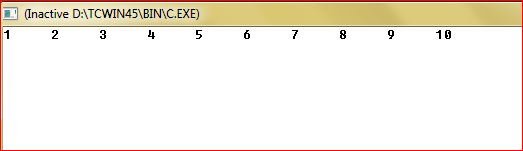
**for(n<10;n++)// هنا لم نعطي قيمة ابتدائية للعداد**

**for(; ;)// هذه الحلقة لانهائية إي تتكرر بدون توقف**

**Q 1: Write a program to print numbers from 1-10 .**

**Solution :**

****

****

**توضيح للسؤال**

**بما انه المطلوب العد من 1 التي تعتبر القيمة الابتدائية إلى 10 التي تعتبر شرط التوقف ويجب إن ينتهي ب(i<=10) بمقدار زيادة 1**

**لو أردنا طباعة الإعداد الفردية ما علينا سوى تغيير مقدار العداد (update) إي زيادة بمقدار 2 (i+=2)**

**رسم توضيحي يبين كيفية عمل for**

القيمة الابتدائية i=1

i++ مقدار الزيادة

طباعة القيمة

**تحقق الشرط**

(i<=10) شرط التوقف

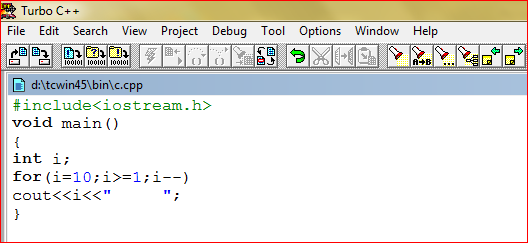
**في حال لم يتحقق**

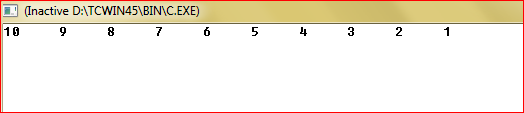
**الشرط**

خروج من جملة for

**Q 1: Write a program to print numbers from 10-1.**

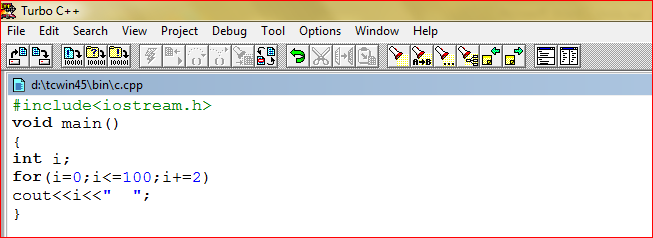
**Solution :**

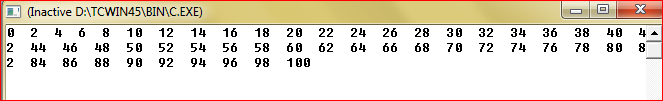
****

****

**Q 2: Write a program collects the even numbers .**

**Solution :**

****

****

**توضيح للبرنامج**

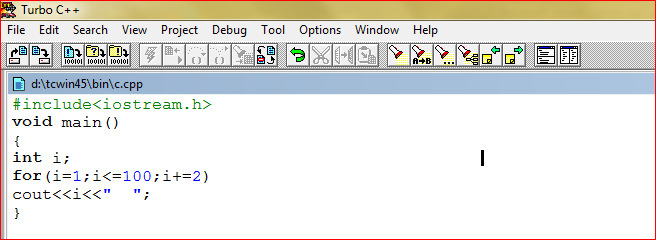
**يبدأ العداد من 0(i=0) ويقارن قيمة i في كل مرة إذا كانت محققة للشرط (i<=100) تزداد قيمة i بمقدار 2 حتى يتم طباعة الإعداد الزوجية**

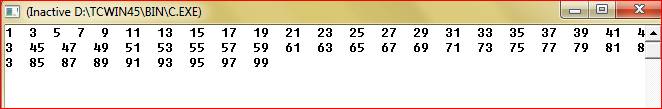
**كل قيمة يقوم بتزويدها 2 يقوم بطباعتها فمثلا في الحالة الأولى (i=0) وهي محققة للشرط (0<=100) وبعدها تزداد بمقدار 2 إي 0+2=2 وهكذا يستمر التكرار إلى إن تصل قيمة i إلى 100 وهي محققة للشرط , إذا هي أخر هي أخر قيمة تطبع وفي الحالة الأخيرة لو زادت قيمة i بمقدار 2 تصبح 102 عندما تدخل شرط المقارنة**

**(102<=100) فان الشرط غير متحقق ,إي تعتبر خارج شرط for ويقوم بطباعة النتائج .**

**Q 2: Write a program collects the odd numbers.**

**Solution :**

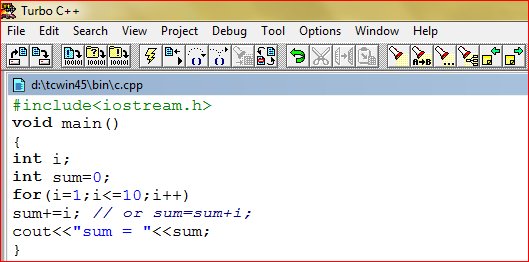
****

****

**في هذا السؤال عرفنا نوع المتغير i داخل for وهذا جائز وهنا يجب الانتباه إلى قيمة i الابتدائية ومقدار الزيادة في مثل هذه الأمثلة إي لو كانت القيمة الابتدائية تبدأ من الصفر لطبع الإعداد الزوجية بدلا عن الفردية**

**Q 1: Write a program to collect numbers from 10-1.**

**Solution :**

****

****

**في هذا السؤال عرفنا متغير sum من نوع int وأعطيناه قيمة ابتدائية وهي 0 لان الصفر هو العنصر المحايد في عملية الجمع ويعتبر sum خزان لجمع الإعداد وألان سوف نقوم بتتبع البرنامج خطوة خطوة لكي نعرف كيف يتم جمع الإعداد**

**when i=1**

**sum=0**

**1<=10 true**

**sum=0+1=1**

**i++**

**i=2**

**sum=1**

**2<=10 true**

**sum=1+2=3**

**i++**

**i=3**

**sum=3**

**3<=10 true**

**sum=3+3=6**

**i++**

**i=4**

**sum=6**

**4<=10 true**

**sum=6+4=10**

**i++**

**i=5**

**sum=10**

**5<=10 true**

**sum=10+5=15**

**i++**

**i=6**

**sum=15**

**6<=10 true**

**sum=15+6=21**

**i++**

**i=7**

**sum=21**

**7<=10 true**

**sum=21+7=28**

**i++**

**i=8**

**sum=28**

**8<=10 true**

**sum=28+8=36**

**i++**

**i=9**

**sum=36**

**9<=10 true**

**sum=36+9=45**

**i++**

**i=10**

**sum=45**

**10<=10 true**

**sum=45+10=55**

**i=11**

**11<=10 false**

**يخرج من for ويطبع الناتج وهو 55**

**الحلقات التكرارية المتداخلة باستخدام Nested For**

**الحلقات المتداخلة:-عبارة عن تكرارات متداخلة مع بعضها وعند تنفيذ جمل معينه يجب إن تحقق شروط الحلقات المتداخلة لكي يتم تنفيذ هذه الجمل تشبه فكره الحلقات المتداخلة فكرة عمل عقارب الساعة فنجد عقرب الثواني يدور 60 دوره ومن ثم يدور عقرب الدقائق بمقدار دقيقه ومن ثم يدور عقرب الدقائق 60 دورة لكي يدور عقرب الساعات مره واحده وهكذا يستمر عقارب الساعة بالدوران .**

**كيفية كتابة الصيغ العامة للتكرارات FOR**

**الحلقة الخارجية وهي بمثابة عقرب الساعاتFOR ( )**

**{**

**الحلقة الوسطى وهي بمثابة عقرب الدقائق FOR ( )**

**{**

**الحلقة الوسطى وهي بمثابة عقرب الثواني FOR ( )**

**الجمل المراد تكرارها حسب الحلقات الثلاث Statements**

**لو نلاحظ كيف تتم حركة عقارب الساعة حيث إن عندما يدور عقرب الثواني نلاحظ إن عقرب الدقائق أيضا يدور لكن ليس بنفس السرعة وكذلك الحال بالنسبة لعقرب الساعات فسرعه العقارب الثلاث مختلفة كذلك الحال بالنسبة للحلقات التكرارية فأن تنفيذها لا يتم مع بعض أو أول من تنفذ هي الحلقة التكرارية الداخلية ثم الوسطى ثم الخارجية وللتوضيح أكثر نأخذ المثال التالي.**

**for(int i=0;i<4;i++)**

**{**

**for(int j=0;j<3;j++)**

**{**

**for(int k=0;k<2;k++)**

**Statement;**

**}}**

**نأخذ القيمة الابتدائية لـ (i=0) نلاحظ أنها محققة للشرط ,وينتقل compiler بالتنفيذ إلى السطر الثاني حيث نلاحظ أيضا وجود حلقة تكراريه أيضا يأخذ القيمة (j=0) ويتأكد من أنها متحققة للشرط وينتقل بالتنفيذ إلى العملية التي بعدها وأيضا يلاحظ وجود حلقه تكرارية يقوم بتنفيذها إلى إن يتوقف شرط الجملة الأخيرة يطبع النتيجة ويعود إلى الحلقة الخارجية من جديد ويقوم بزيادة "i" ومن ثم ينتقل إلى الحلقة الوسطى وأيضا يقوم**

**بزيادة قيمة "j" وينتقل إلى الحلقة الداخلية ويقوم بتنفيذها بالكامل ثم يطبع النتيجة وفي كل مرة ينفذ حيث نلاحظ إن أول حلقة ينتهي تنفيذها بالكامل وهي الحلقة الداخلية ومن ثم الحلقة الوسطى ثم الخارجية وبعدها يعرض النتائج المطلوبة.**

I=0

I < 4

**في**

**حال حال**

J = 0

**عدم**

J < 3

**تحقق**

K = 0

**الشرط**

K < 2

**في**

طباعة القيمة

**حال حال**

**عدم عدم**

K ++

**تحقق**

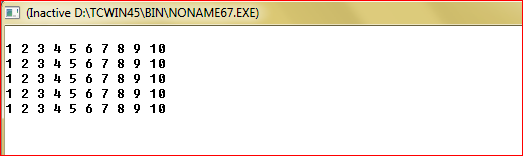
J++

**الشرط**

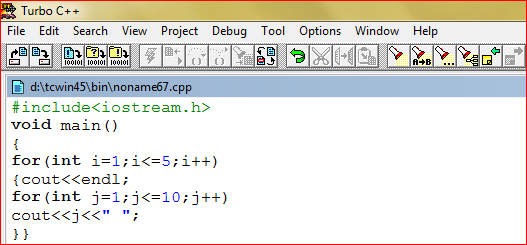
I++

طباعة الناتج

**Q1 : Write a program performs the following figure .**

****

**Solution :**

****

**عندما تكون (i=0)فتنفذ الدورة الثانية أو الحلقة التكرارية الثانية بالكامل وتعرض النتيجة ومن ثم تزداد قيمة (i) بقيمة " 1" i=1 أيضا تنفذ الحلقة التكرارية الثانية بالكامل وتطبع النتيجة وهكذا حيث انه في كل مره تزداد قيمة (i)تنفذ الحلقة التكرارية الثانية بالكامل إلى إن يتوقف شرط الحلقة التكرارية الأولى إي إن**

**1 2 3 4 5 6 7 8 9 10 عندما i=0**

**1 2 3 4 5 6 7 8 9 10 عندما i=1**

**1 2 3 4 5 6 7 8 9 10 عندما i=2**

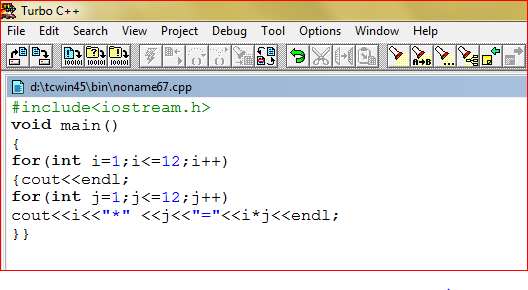
**1 2 3 4 5 6 7 8 9 10 عندما i=3**

**1 2 3 4 5 6 7 8 9 10 عندما i=4**

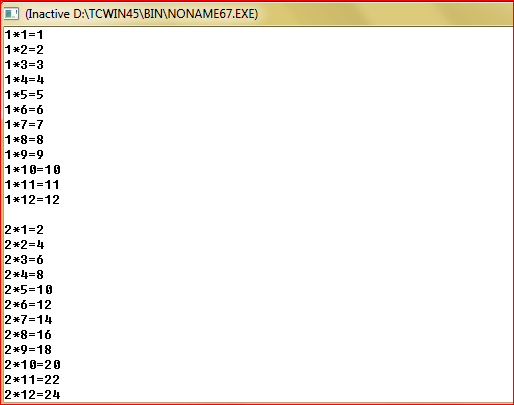
**1 2 3 4 5 6 7 8 9 10 عندما i=5**

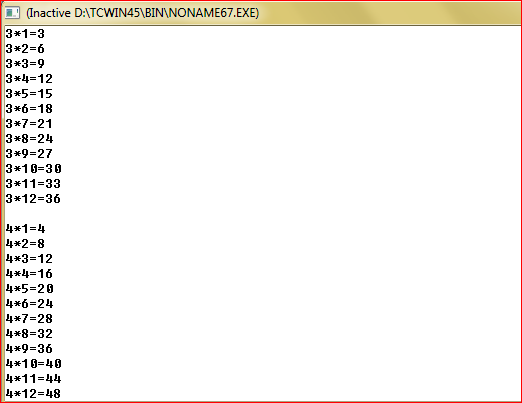
**Q2 : Write a program prints the multiplication table .**

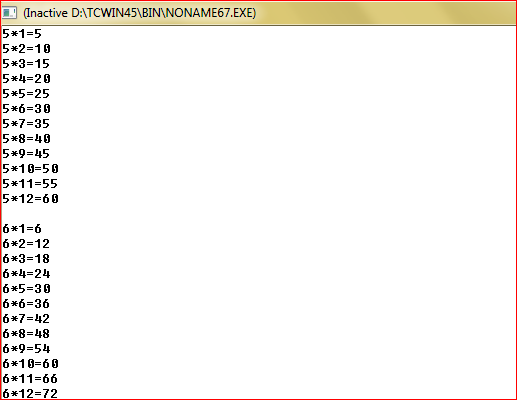
**Solution :**

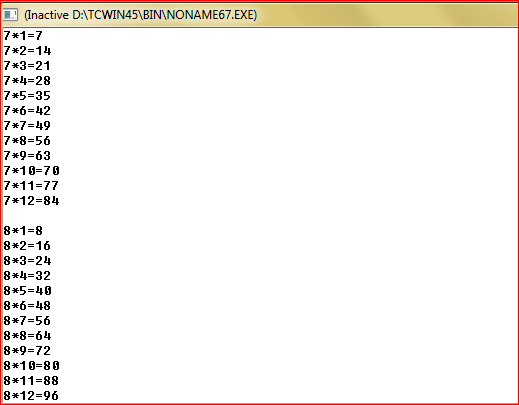
****

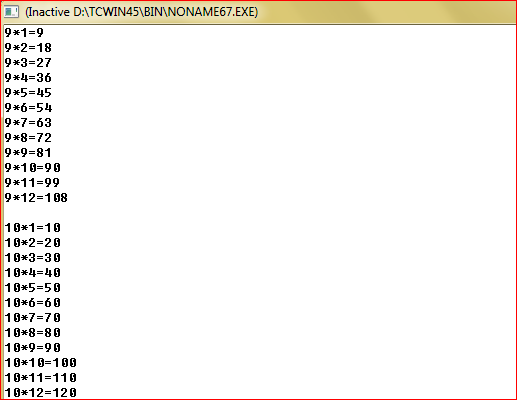
**Note:- عند تنفيذ البرنامج لا تكتب الحروف الكبيرة في اللغة الإنجليزية وإنما نكتبها بالحروف الصغيرة عند التنفيذ.**

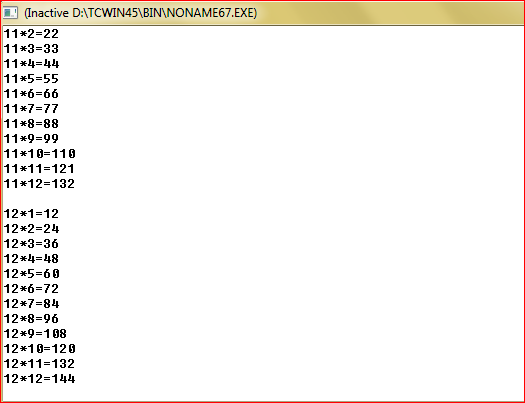
****

****

****

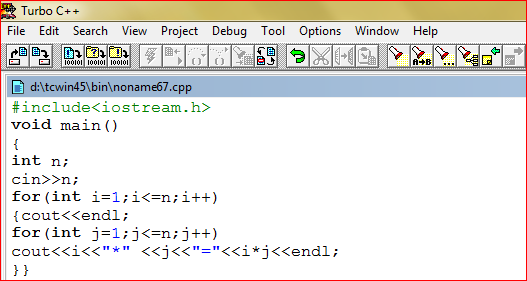
****

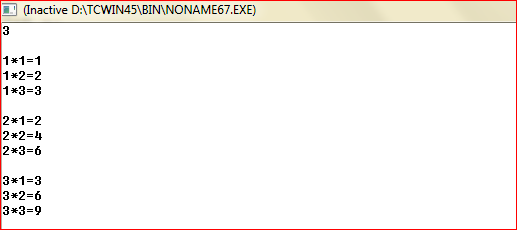
****

****

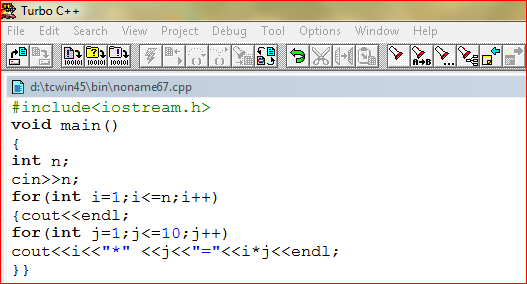
**Q3 : Write a program to print the multiplication table to the any number.**

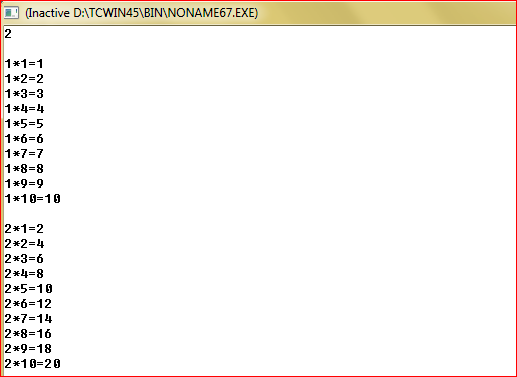
**Solution :**

****

****

**إذا قمنا بتغير قيمة j في السؤال فأنه بدل أن يطبع كل جدول إلى 3 سوف يقوم بطباعة كل جدول إلى قيمة j**

****

****

**Q 4 : Write a program to calculate the factorial of the any number .**

**الجواب:-**

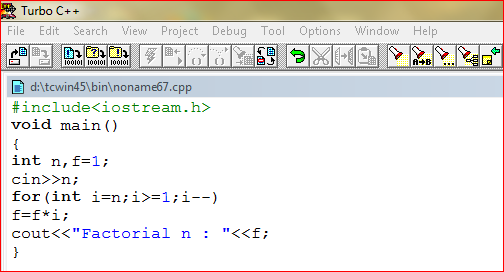
**إن فكرة المضروب هي كالأتي:**

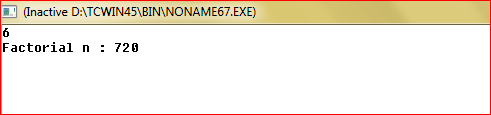
**مثلا لو كان لدينا العدد 5 ونريد إن نحسب المضروب لهذا العدد ونقصد( !(5**

**5!=5x4x3x2x1=120**

**6!=6x5x4x3x2x1=**

**وهكذا حيث إن العدد يضرب مع سابقة المتناقص بمقدار واحد في كل مره إلى إن يصل 1 ويتوقف**

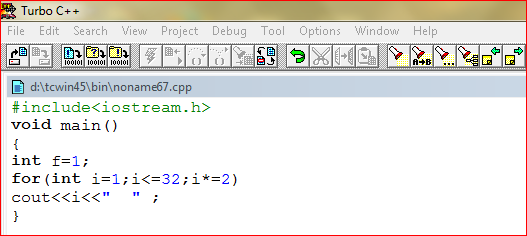
****

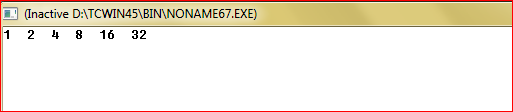
****

**Q5 : Write a program to print the series .**

**1 2 4 8 16 32**

**Solution :**

****

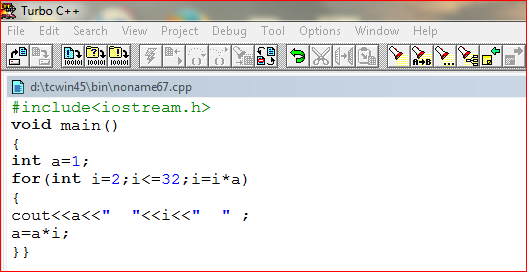
****

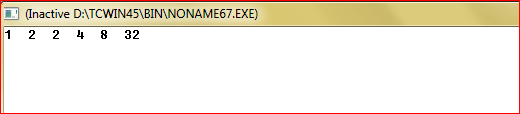
**لو لاحظنا المتسلسلة نجد انه عند ضرب كل عدد 2\* يعطي العدد الذي بعده إي عند ضرب 1x2=2 وعند ضرب 2x2=4 وعند ضرب 4x2=8 وهكذا .**

**Q6 : Write a program to print the series .**

**1 2 2 4 8 32**

**Solution :**

****

****

**أسلوب التكرار باستخدام while &do\_ while**

1. **إما الصيغة العامة للــــــ while**

**تعني مادام الشرط متحقق تنفذ التعليمة الثانية while(condition)**

**Statements;**

1. **الصيغة العامة للـــــ do- while**

**تنفذ التعليمة التالية مادام الشرط متحققا do**

**Statements;**

**while(condition)**

**نلاحظ إن أسلوب التكرار باستخدام while &do\_ while يختلف عن for من حيث:-**

* **1-عند استخدام while &do\_ while يجب إن نضع شرطا قابلا للتحقق لكي نضمن انتهاء الحلقة التكرارية لأنه إذا كان الشرط غير متحقق يعني الدخول في الحلقة النهائية وبالتالي يتوقف عن العمل.**
* **2-إن while &do\_ whileلا تمتلكان عدادات زيادة أو نقصان في داخلهما لذلك فهما تقبلان الشروط المنطقية والعددية لكن بإمكان المستخدم إن يضع لهما عداد سنتعرف على هيكلتهما لاحقا.**
* **3- الحلقة التكرارية while نقوم باختيار الشرط أولا ثم تنفذ العمليات التي مداخلهما إذا كان الشرط متحققا .إما do-while فهي تنفذ تعليمة ثم تتحقق من الشرط وإذا تحقق أعادت تكرارات العمليات التي بداخلهما.**

**هيكلية while التي تحتوي على عداد:-**

**Initialization**

**While(condition)**

**{**

**Statements**

**Update;**

**}**

**هيكلية do-while التي تحتوي على عداد:-**

**Initialization**

**Do**

**{**

**Up date**

**Statements**

**}**

**While(condition)**

**الحلقات while المتداخلة Nested While Loop**

**Initialization1;**

**Initialization2;**

**While(condition1)**

**{**

**While(condition2)**

**{**

**Statements;**

**Update2;**

**}**

**Update1;**

**الحلقات do-while المتداخلة Nested do -While Loop**

**Initialization1;**

**Initialization2;**

**Do1;**

**{**

**Statements;**

**Do2;**

**{**

**Statements;**

**Update2;**

**Update1;**

**}**

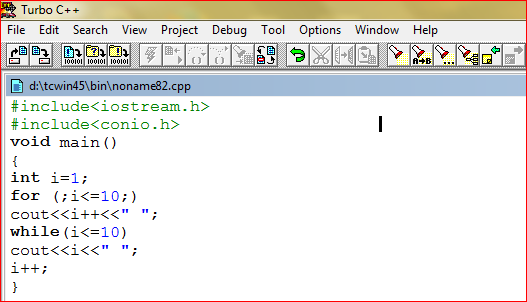
**While(condition1)**

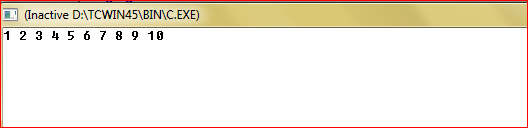
**While(condition2)**

**}**

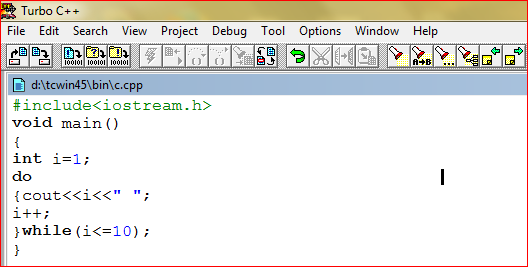
**Q 1: Write a program to print numbers from 1-10 .**

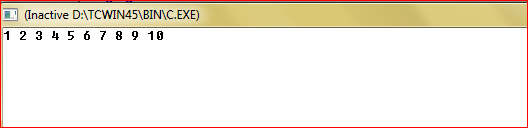
**Solution :**

****

****

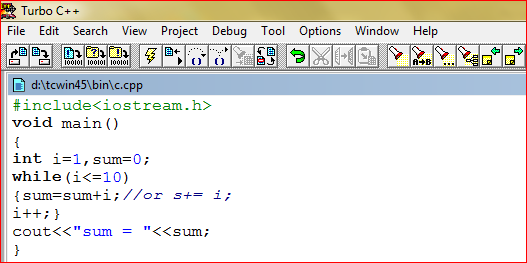
**DO\_ While**

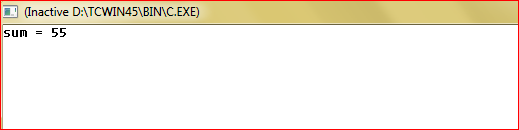
****

****

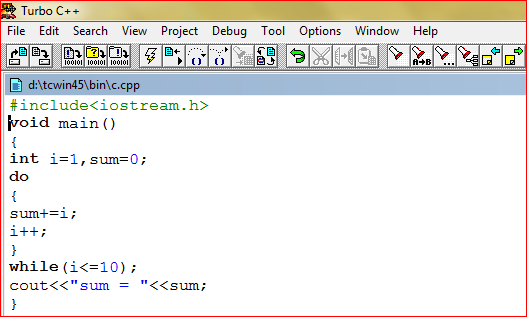
**Q 2: Write a program collects the numbers from 1-10 .**

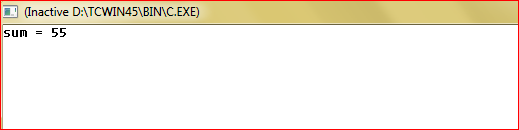
**Solution :**

****

****

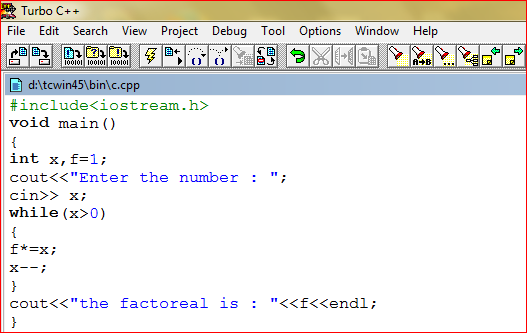
**Do \_While**

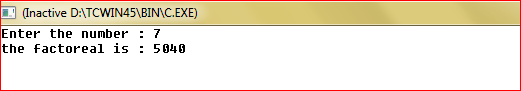
****

****

**Q 3 : Write a program to calculates factorial numbers .**

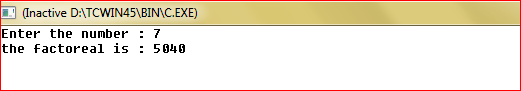
**Solution :**

****

****

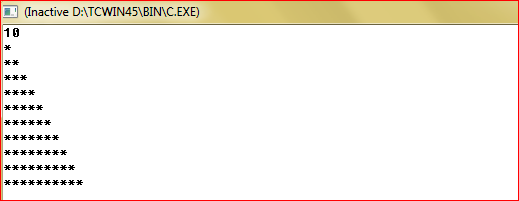
**Do \_ While**

****

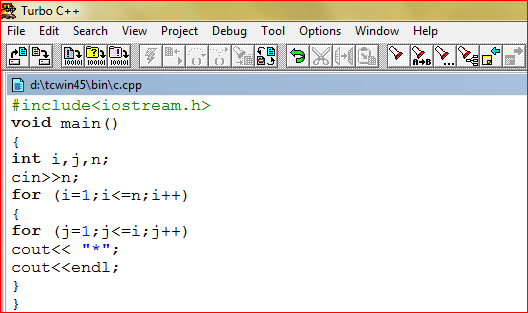
****

**بعض البرامج لطباعة الإشكال :**

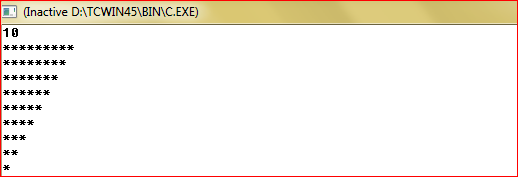
**Q1 : Write a program performs the following figure .**

****

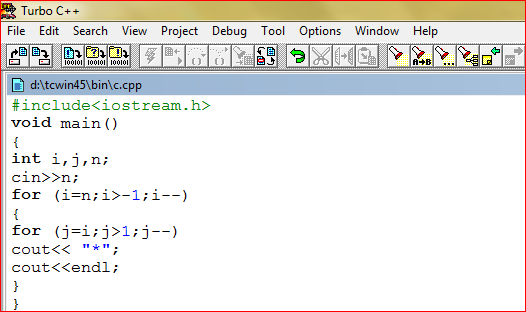
**Solution :**

****

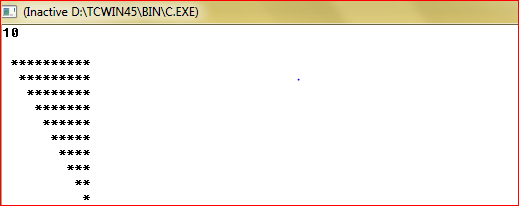
**Q2 : Write a program performs the following figure .**

****

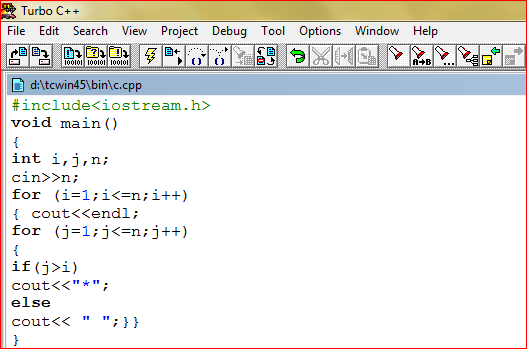
**Solution :**

****

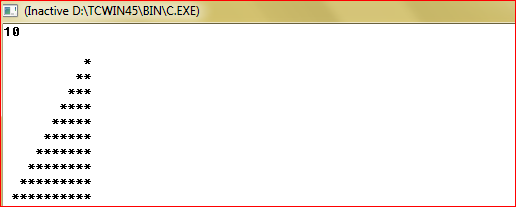
**Q3 : Write a program performs the following figure .**

****

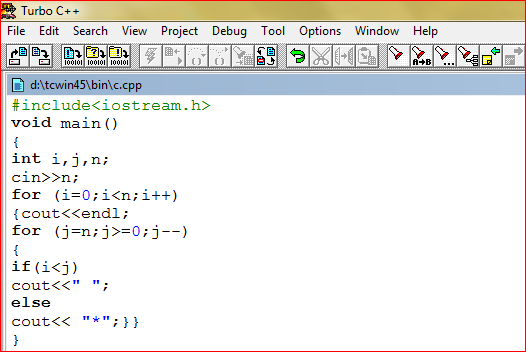
**Solution :**

****

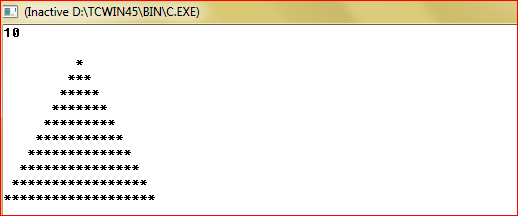
**Q4 : Write a program performs the following figure .**

****

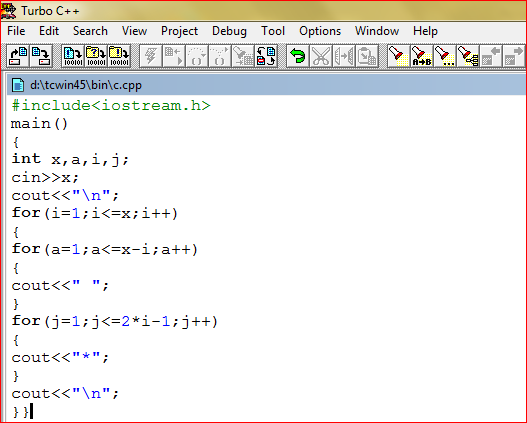
**Solution :**

****

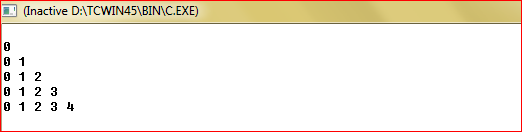
**Q5 : Write a program performs the following figure .**

****

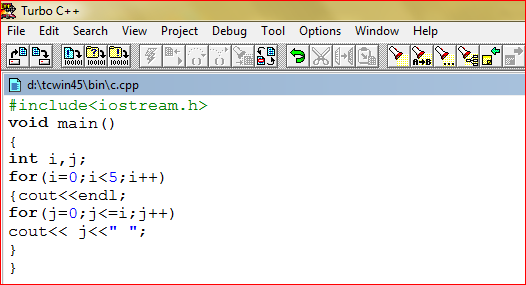
**Solution :**

****

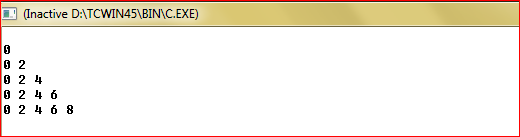
**Q6 : Write a program performs the following figure .**

****

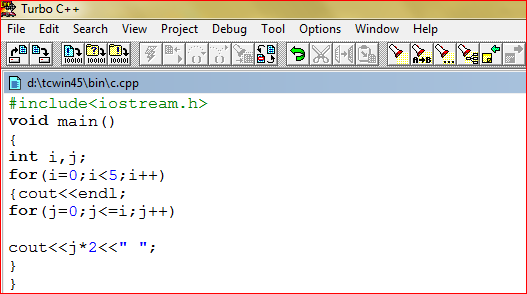
**Solution :**

****

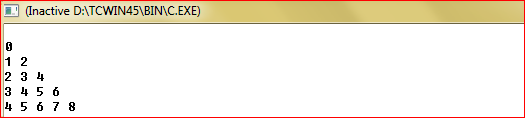
**Q7 : Write a program performs the following figure .**

****

**Solution :**

****

**Q8 : Write a program performs the following figure .**

****

**Solution :**

****

**المصفوفات Matrices or Array :**

**تستخدم المصفوفات في لغة c++ أو إي لغة أخرى لتخزين البيانات ذات الصلة (من نوع واحد أي كل البيانات من نوع أما int or float or char ) مفهرسة تحت أسم متغير واحد .**

**يمكن أن نعتبر المصفوفات متغير يحمل أكثر من قيمة من نفس النوع , وعلى هذا النحو فأن المصفوفات تساعد المبرمج على تنظيم مجموعة من البيانات بكفاءة .**

**كما ويمكن استعمال المصفوفات مع الدوال سيأتي شرحها في هذا الفصل ,فائدة المصفوفات كبيرة وطرق استعمالها كثيرة ومتنوعة وأن طرقة التعامل مع المصفوفات مشابه لطرقة التعامل مع المتغيرات في كما ذكرنا عبارة عن متغير يحمل أكثر من قيمة وله DataType أيضاً .**

**8 7 6 5 4 3 2 1 0**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **I** | **H** | **G** | **F** | **E** | **D** | **C** | **B** | **A** |

**Array char**

**حيث أن :**

**Char Array هو اسم مصفوفة من نوع بيانات char .**

**A,B,C,… هي البيانات التي يحملها المتغير Array .**

**0,1,2,3,… هي index المصفوفة حيث أن لكل خلية أو موقع يحتوي على بيانات له فهرسة (index) يمكن الوصول إلى موقع أو أي رمز من خلال معرفة فهرسة الموقع المراد الوصول إليه . ولابد من الإشارة إلى إن فهرسة المصفوفة دائما تبدأ من 0 .**

**الإعلان عن المصفوفات .**

**DataType NameArray [Size] ;**

**حيث أن:**

**DataType نوع البيانات .**

**NameArray اسم المصفوفة .**

**Size هو حجم المصفوفة .**

**أنضر إلى الأمثلة التالية :**

**Int X[5] ; تعني مصفوفة أسمها X من موع بيانات int حجمها 5 .**

**Float Y[10] ; تعني مصفوفة أسمها Y من نوع بيانات float حجمها 10 .**

**Char Z[8]; تعني مصفوفة اسمها Z من نوع بيانات char حجمها 8 .**

**أنواع المصفوفات .**

**هناك ثلاث أنواع من المصفوفات وهي مصفوفات أحادية الأبعاد و مصفوفات ثنائية الأبعاد و مصفوفات متعددة الأبعاد في هذه الكتاب سنتطرق لشرح المصفوفات ذات البعد الواحد والمصفوفات ذات البعدين .**

* **المصفوفة ذات البعد الواحد.**

**هي مصفوفة تحتوي على صف واحد من البيانات .**

**8 7 6 5 4 3 2 1 0**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **I** | **H** | **G** | **F** | **E** | **D** | **C** | **B** | **A** |

**طريقة الإعلان عن المصفوفة ذات البعد الواحد .**

**DataType NameArray [size];**

**Example : int x[12];**

* **المصفوفة ذات البعدين .**

**هي المصفوفة التي تحتوي على عدد من الصفوف والأعمدة التي تحمل البيانات .**

|  |
| --- |
| **0,0** |
| **0,1** |
| **0,2** |
| **1,0** |
| **1,1** |
| **1,2** |
| **2,0** |
| **2,1** |
| **2,2** |

|  |  |  |
| --- | --- | --- |
| **0,2** | **0,1** | **0,0** |
| **1,2** | **1,1** | **1,0** |
| **2,2** | **2,0** | **2,0** |

**مخطط يوضح طريقة خزن عناصر مصفوفة ثنائية في الذاكرة**

**الإعلان عن المصفوفة الثانية**

**DataType NameArray [Row][Colume] ;**

**Example : int x[3][3];**

**تهيئة المصفوفات .**

**تهيئة المصفوفة الأحادية .**

**يقصد بتهيئة المصفوفة هو إعطاء قيم للمصفوفة حيث يوجد طريقتين لإعطاء قيم للمصفوفة .**

**الطريقة الأولى : الطريقة المباشرة وهي بعد كتابة نوع البيانات واسم المصفوفة يتم إدخال البيانات والتي تكون داخل قوسين المجموعة .**

**Example : int x [3]={ 1, 5, 2};**

**أي أن العنصر الأول هو : x[0]=1 ;**

**والعنصر الثاني هو :x[2]=5 ;**

**والعنصر الثالث هو : x[2]=2 ;**

**الطريقة الثانية : الطريقة الغير مباشرة وهي إدخال القيم عن طرق cin>> .**

**Cin>>x[0];**

**Cin>>x[1];**

**Cin>>x[2];**

**حيث أن هذه الطريقة اقل كفاءة لأنه لو كانت لدينا مصفوفة مكونة من 100 رقم فهل من الطبيعي استخدام إيعاز cin>> مئة مرة ! لهذا السبب نلجئ إلى استخدام loop في إدخال القيم للمصفوفة .**

**Int x[5];**

**For(int i=0;i<5;i++)**

**Cin>>x[i];**

**تهيئة المصفوفة الثنائية .**

**بنفس الطر ق السابقة يمكن إعطاء قيم للمصفوفة الثنائية .**

**الطريقة الأولى:**

**int x[3][3]={1,2,3,4,5,6,7,8,9};**

**أي أن :**

**X[0][0]=1;**

**X[0][1]=2;**

**X[0][]2]=3;**

**X[1][0]=4;**

**X[1][1]=5;**

**X[1][2]=6;**

**X[2][0]=7;**

**X[2][1]=8;**

**X[2][2]=9 ;**

**(0,0) (0,1) (0,2) القطر الثانوي**

**(1,0) (1,1) (1,2)**

**(2,0) (2,1) (2,2) القطر الرئيسي**

**(0,0) (0,1) (0,2) عناصر فوق القطر الرئيسي**

**(1,0) (1,1) (1,2)**

**عناصر تحت القطر الرئيسي (2,0) (2,1) (2,2)**

**عناصر فوق القطر الثانوي (0,2) (0,1) (0,0)**

**(1,0) (1,1) (1,2)**

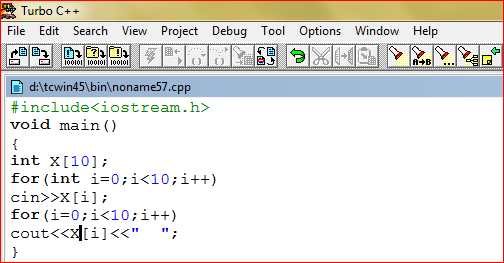
**(2,0) (2,1) (2,2) عناصر تحت القطر الثانوي**

* **عند طباعة عناصر القطر الرئيسي فأنه يجب أن تكون قيمة الصف تساوي قيمة العمود أي . ( i==j )**
* **عند طباعة عناصر القطر الثانوي فأنه يجب أن نجمع قيمة الصف مع العمود ونساويها لحجم المصفوفة ناقص 1 , أي (i+j==size-1) .**
* **عند طباعة عناصر فوق القطر الرئيسي فأنه يجب أن تكون قيمة الصف أصغر من قيمة العمود أي (i < j) .**
* **عند طباعة عناصر أسفل القطر الرئيسي فأنه يجب أن تكون قيمة الصف أكبر من قيمة العمود أي ( i > j ).**
* **عند طباعة عناصر أعلى القطر الثانوي فأنه يجب أن يكون جمع قيمة الصف مع العمود أصغر من حجم المصفوفة ناقص 1 أي ( i+j < size -1) .**
* **عند طباعة عناصر أسفل القطر الثانوي فيجب أن يكون جمع قيمة الصف مع العمود أكبر من حجم المصفوفة ناقص 1 أي (i+j > size-1 ) .**

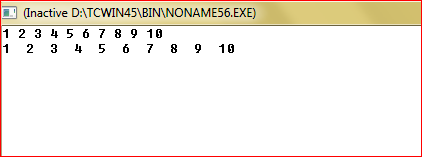
**وألان نأخذ بعض الأمثلة على المصفوفات الأحادية و الثنائية .**

**Q1/ Write a program asks the user to enter ten elements of the 1D array and then print .**

**Solution :**

****

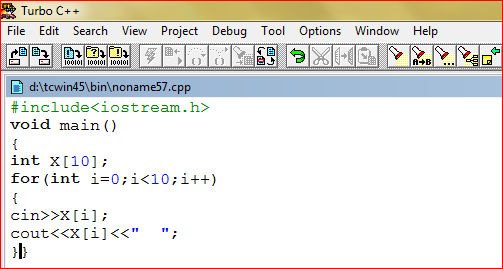
**Execution screen .**

****

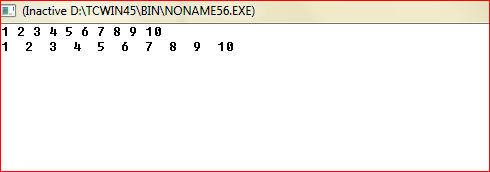
**Note : لدينا مصفوفة مكونة من 10 عناصر لكي ندخل قيم لهذه المصفوفة نحتاج إلى أسلوب التكرار لكي يتم أدخل عشرة عناصر وأيضاً مطلوب في السؤال طباعتها أيضاً نحتاج إلى كتابة أسلوب التكرار مرة ثانية لكي يتم عرض عشرة عناصر .**

**أن for الأولى هي للمعالجة و for الثانية هي للطباعة لكي يتم طباعة عشرة عناصر حيث انه عند حذف for الثانية جيب وضع أقواس بعد جملة for حتى يتم تنفيذ البرنامج بالشكل الصحيح .**

**نفس البرنامج السابق باستخدام for واحدة .**

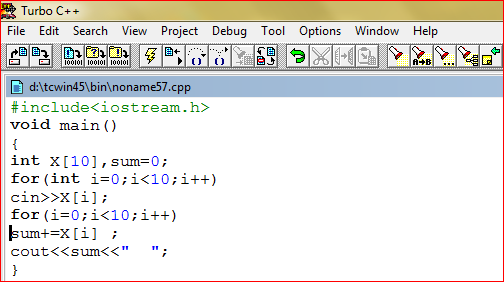
****

**Execution screen .**

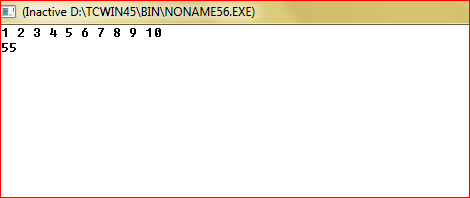
****

**Q2: Write a program that collects elements array 1D .**

**Solution:**

****

**Execution screen .**

****

**مخطط توضيحي للبرنامج :**

**i = 0**

**في حال تحقق الشرط**

**i < 10**

**في حال عدم تحقق الشرط**

**cin>> X[i]**

**i++**

**i < 10**

**i=0**

**في حال عدم تحقق الشرط**

**sum = 0**

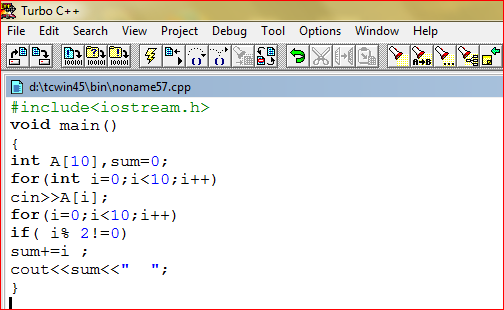
**cout<<sum**

**i++**

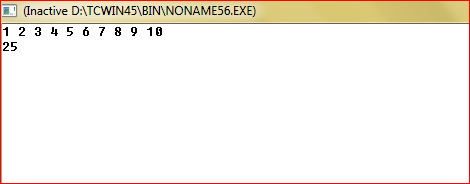
**sum+=X[i]**

**Q3: Write a program that collects odd numbers in 1D array .**

**Solution :**

****

**Execution screen .**

****

**مخطط توضيحي للبرنامج بعد إدخال البيانات أي جزء المعالجة :**

**i=0**

**cout<< sum**

**i++**

**sum +=i**

**If ( I % 2 != 0 )**

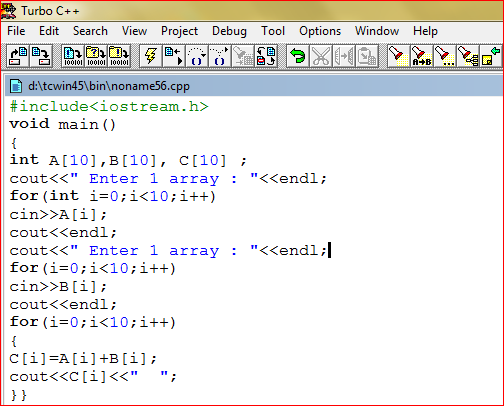
**i < 10**

**في حال عدم تحقق الشرط**

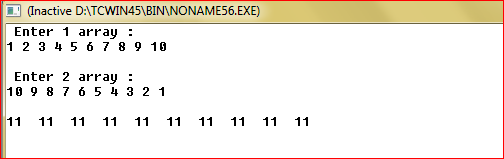
**في حال عدم تحقق الشرط**

**Q4 : Write a program that collect two arrays 1D .**

**Solution :**

****

**Execution screen .**

****

**إدخال المصفوفة الأولى**

**i=0**

**i++**

**Cin>>B[i]**

**i < 10**

**i=0**

**i++**

**Cin >> A[i]**

**i<10**

**i=0**

**i < 10**

**في حال عدم في حال**

**تحقق الشرط تحقق**

**C[i]=A[i]+B[i]**

**الشرط**

**Cout<<C[i]<<" "**

**i++**

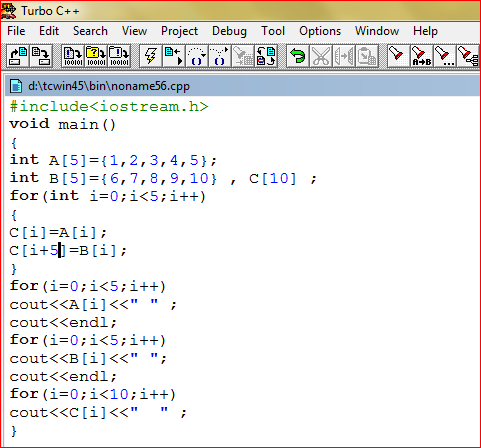
**إدخال المصفوفة الثانية**

**الخروج من for**

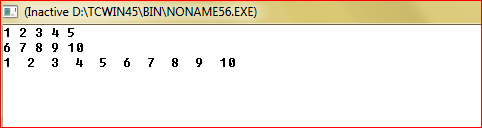
**في حال عدم تحقق الشرط**

**Q5: Write a program that integrates two matrices each them contain five elements .**

**Solution :**

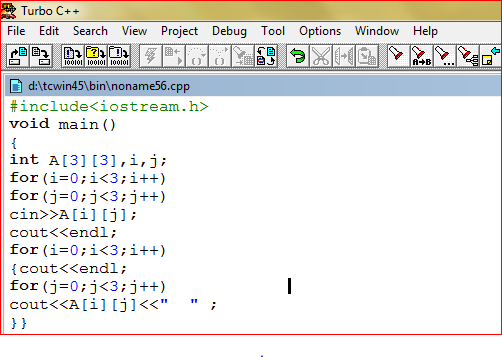
****

**Execution screen .**

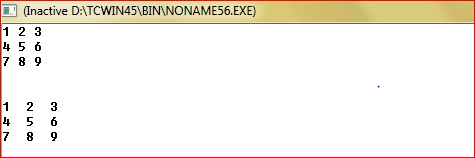
****

**Q6: Write a program enter the array and print them ,assume the size of array 3x3 .**

**Solution :**

****

**Execution screen .**

****

**مخطط يوضح كيفية إدخال عناصر مصفوفة ثنائية وطباعتها :**

**i++**

**j++**

**Cout<<A[i][j]**

**j < 3**

**j = 0**

**Cout<<endl**

**i < 3**

**i =0**

**i++**

**j++**

**Cin>>A[i][j]**

**j < 3**

**j = 0**

**i < 3**

**i =0**

**في حال**

**عدم تحقق**

**الشرط**

**في في حال**

**عدم تحقق**

**الشرط**

**في حال**

**عدم تحقق**

**الشرط**

**في حال**

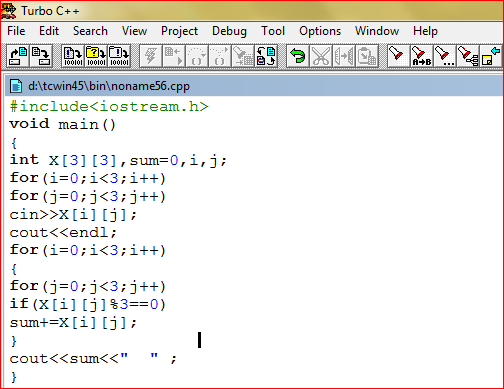
**عدم تحقق**

**الشرط**

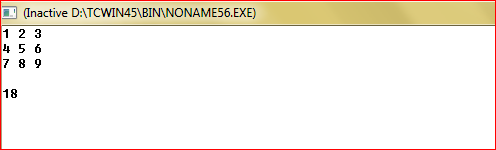
**الخروج من for**

**Q7: Write a program enter the array , and collects the number divisible by 3 ,assume the size of array 3x3.**

**Solution :**

****

**Execution screen .**

****

**مخطط يوضح كيفية جمع الإعداد التي تقبل القسمة على 3 في مصفوفة ثنائية بعد إدخال البيانات :**

**Cout<<sum**

**i++**

**j++**

**Sum+=X[i][j]**

**X[i][j]%3==0**

**j < 3**

**j=0**

**i < 3**

**i=0**

**في حال**

**عدم تحقق**

**الشرط**

**في حال**

**عدم تحقق**

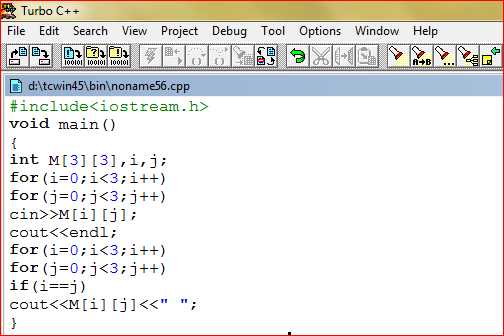
**في حال الشرط**

**عدم تحقق**

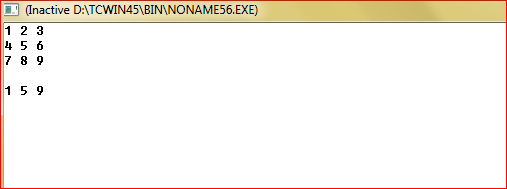
**الشرط**

**Q8: Write a program to print the elements the main diameter.**

**Solution :**

****

**Execution screen .**

****

**مخطط يوضح كيفية طباعة عناصر القطر الرئيسي بعد إدخال البيانات :**

**خروج من for**

**i++**

**j++**

**Cout<<M[i][j]**

**If (i==j)**

**j < 3**

**j = 0**

**i < 3**

**i = 0**

**في حال**

**عدم تحقق**

**الشرط**

**في**

**في حال حال**

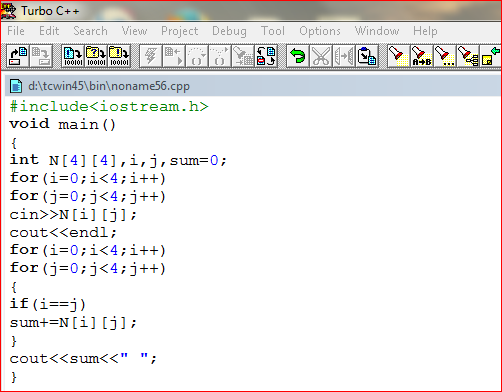
**عدم تحقق عدم**

**الشرط تحقق**

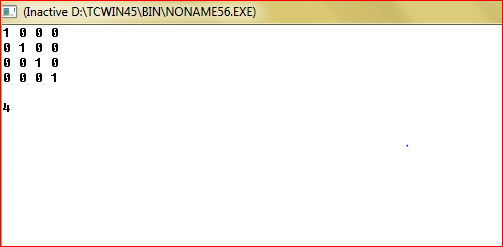
**الشرط**

**Q 9: Write a program that calculates the sum of the main diameter elements in 2D array , the size of array 4x4.**

**Solution :**

****

**Execution screen .**

****

**مخطط يوضح كيفية جمع عناصر القطر الرئيسي في مصفوفة ثنائية بعد إدخال البيانات :**

**i++**

**j++**

**Cout<<sum**

**j < 4**

**j=0**

**i < 4**

**Sum+= N[i][j]**

**If (j==i)**

**i=0**

**في حال**

**عدم تحقق**

**الشرط**

**في حال**

**عدم تحقق**

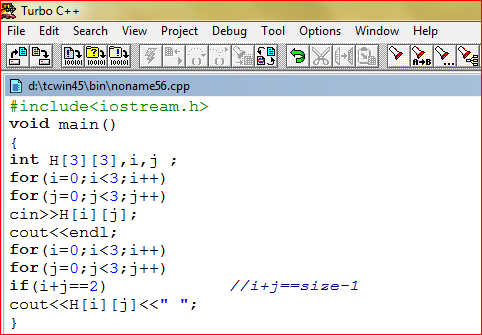
**في حال الشرط**

**عدم تحقق**

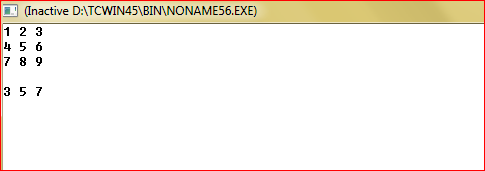
**الشرط**

**Q10 : Write a program to print the elements the secondary diameter.**

**Solution:**

****

**Execution screen .**

****

**مخطط يوضح كيفية طباعة عناصر القطر الثانوي بعد إدخال البيانات :**

**Cout<<H[i][j]**

**i++**

**j++**

**Cout<<H[i][j]**

**If(i+j==2)**

**j < 3**

**j =0**

**i < 3**

**i=0**

**في حال**

**عدم تحقق**

**الشرط**

**في حال**

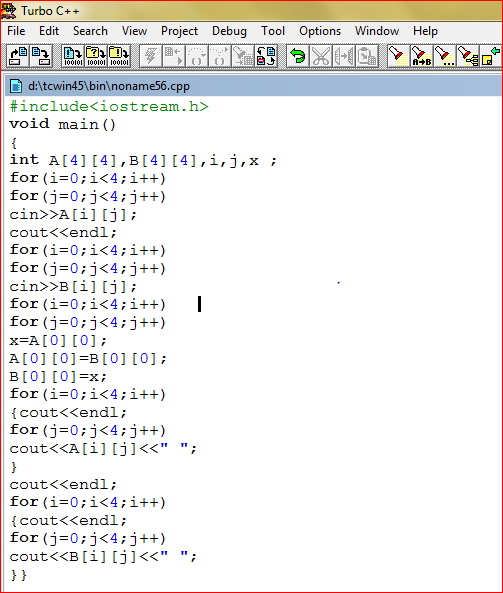
**في حال عدم تحقق**

**عدم تحقق الشرط**

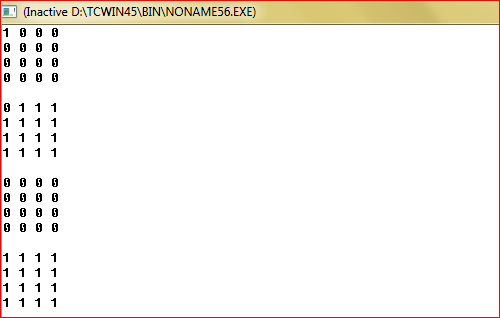
**الشرط**

**Q11 : Write a program enter two array then the first element of the first array switches to where the first element of second array ,assume the size of arrays 4x4 .**

**Solution :**

****

**Execution screen .**

****

**مخطط يوضح طريقة تبديل العنصر الأول في المصفوفة الأولى بالعنصر الأول في المصفوفة الثانية :**

**i = 0**

**i < 4**

**في حال**

**خروج من for**

**i++**

**j++**

**X=A[0][0]**

**A[0][0]=B[0][0]**

**B[0][0]=X**

**j < 4**

**j = 0**

**عدم تحقق**

**الشرط**

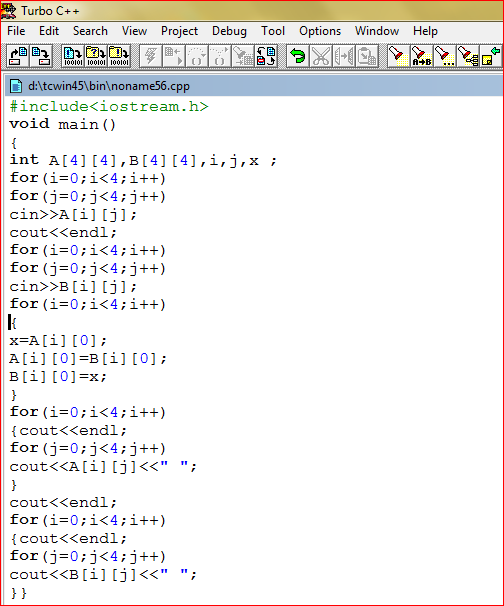
**في حال**

**عدم تحقق**

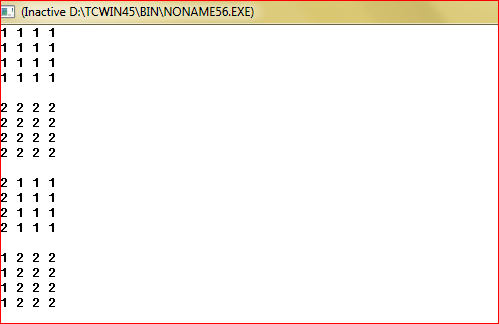
**الشرط**

**Q12 : : Write a program that enter two array then the first row of the first array switches to where the first row of second array ,assume the size of arrays 4x4 .**

**Solution:**

****

**Execution screen .**

****

**مخطط يوضح كيفية تبديل عناصر العمود الأول من المصفوفة الأولى بالعمود الأول من المصفوفة الثانية :**

**i=0**

**i < 4**

**في حال**

**j = 0**

**عدم تحقق**

**الشرط**

**X=A[i][0]**

**A[i][0]=B[i][0]**

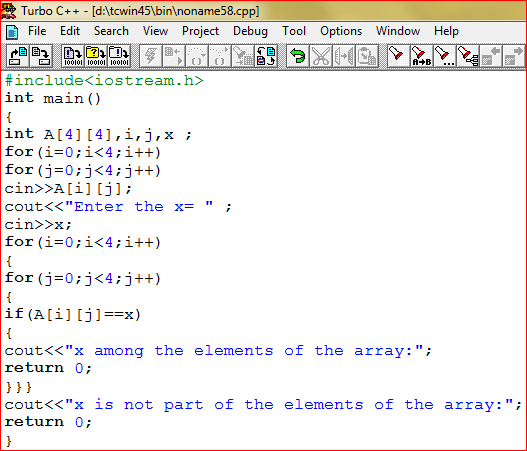
**B[i][0]=x**

**i++**

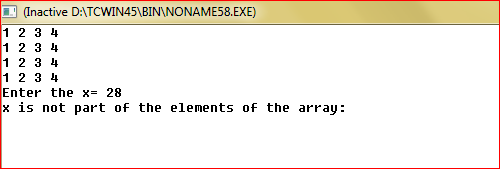
**الخروج من for**

**Q13 : Write a program enters array , then the user enters the number and looking if number from within the array or not , assume the size of array 4x4 .**

**Solution :**

****

**Execution screen .**

****

**مخطط توضيحي للسؤال :**

**i=0**

**i<4**

**في حال**

**j=0**

**عدم تحقق**

**الشرط**

**Cout<<"x is not part of the elements of the array "**

**Cout<<"x among the elements of the array "**

**الخروج من for**

**i++**

**j++**

**If(A[i][j]==x )**

**j<4**

**j=0**

**i<4**

**i=0**

**i++**

**j++**

**Cin>>A[i][j]**

**j<4**

**في حال**

**عدم تحقق**

**الشرط**

**في حال**

**عدم**

**تحقق**

**في الشرط**

**حال**

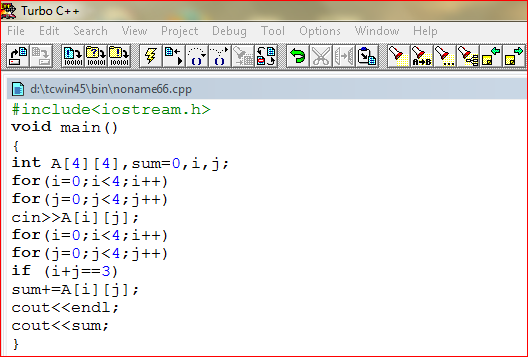
**في حال عدم**

**عدم تحقق تحقق**

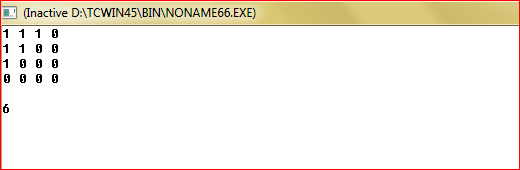
**الشرط الشرط**

**Q14 : Write a program that collects the elements above the secondary diameter .**

**Solution :**

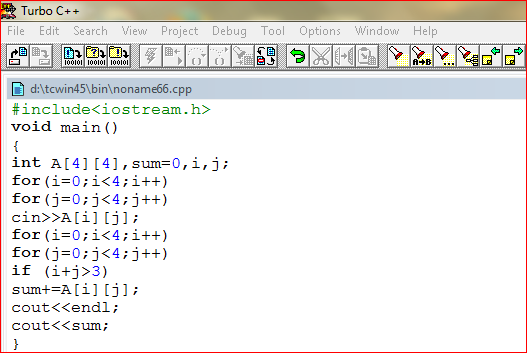
****

**Execution screen .**

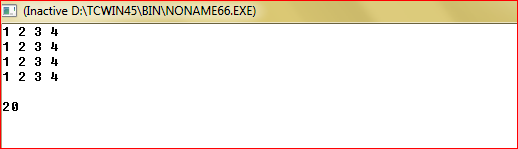
****

**Q15 : Write a program that collects the elements under the secondary diameter .**

**Solution :**

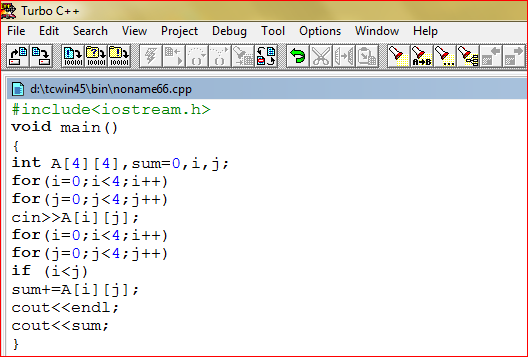
****

**Execution screen .**

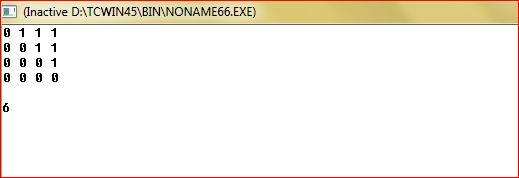
****

**Q16: Write a program that collects the elements above the main diameter .**

**Solution :**

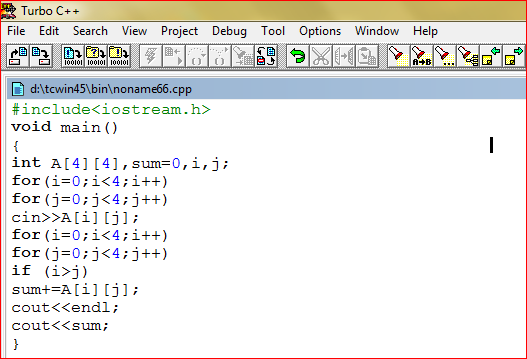
****

**Execution screen .**

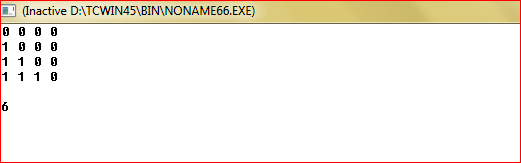
****

**Q17: Write a program that collects the elements under the main diameter .**

**Solution :**

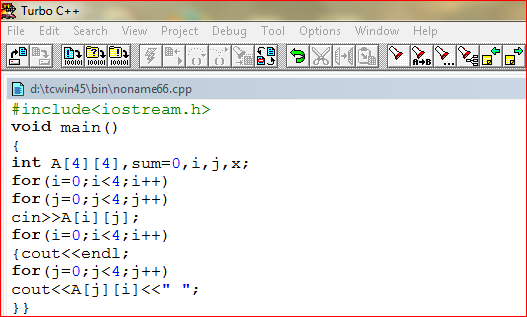
****

**Execution screen .**

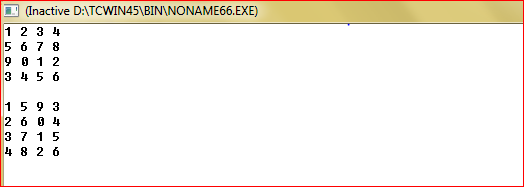
****

**Q18 : Write a program to swap the rows in the columns .**

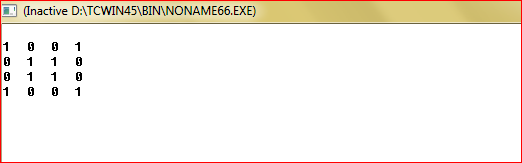
**Solution :**

****

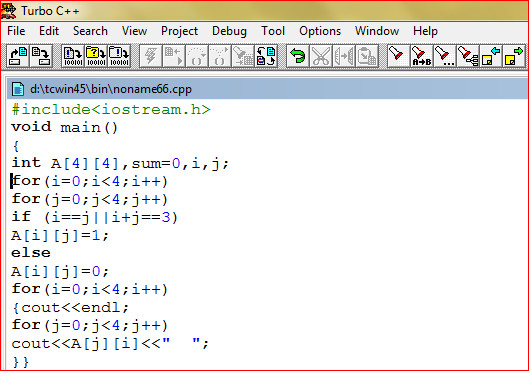
**Execution screen .**

****

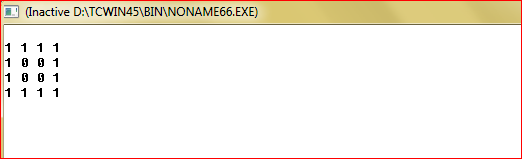
**Q19 : Write a program performs the following figure .**

****

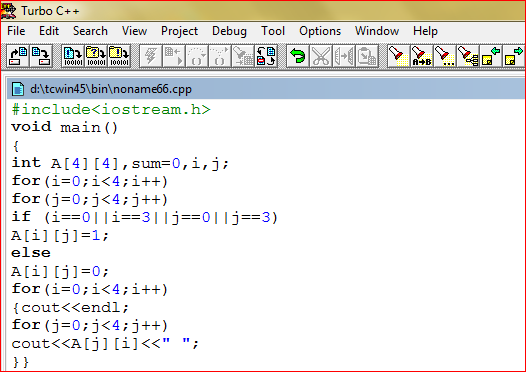
**Solution :**

****

**Q20 : Write a program performs the following figure .**

****

**Solution :**

****

**الدوالfunction**

**إن استخدام الدوال يسهل إلى حد كبير عملية البرمجة ويوفر المساحات المستخدمة في الذاكرة وتلافي التكرار في خطوات البرنامج التي تتطلب عملاً مشابهاً لعمل تلك الدوال يكتفي باِستدعائها باِسمها للقيام بالعمل المطلوب**

**تتكون الدوال من ثلاث أجزاء:-**

**الجزء الأول هو جزء الإعلان عن الدوال:**

**قبل اِستدعاء أي دالة يجب الإعلان عنها وهذا الإعلان يتضمن اسم الدالة وعدد متغيراتها و أنواعها ونوع القيمة المعادة من خلال الدالة وينتهي بفاصلة منقوطة**

**مثال على الإعلان عن دالة ما :-**

**Int print(int);**

**حيث إن:-**

**-النوع-:int الموجود بين قوسين يدل على نوع البيانات أو المتغيرات التي يتم تمريرها إلى الدالة.**

**--: printهو اسم الدالة.**

**-النوع intger:- الذي يسبق اسم الدالة يدل على القيمة التي ترجعها الدالة.**

**-مع ملاحظة إن الإعلان ينتهي بفاصلة منقوطة.**

**الجزء الثاني:-جزء التعريف function Definition:-**

**ويأخذ الصيغة الآتية.**

**Return\_valueType function\_name(parameters)**

**{**

**Statements**

**}**

**حيث أن:-**

**Return\_valueType-:-تمثل نوع القيمة المعادة التي أي التي ترجعها الدالة .**

**Function name-:-وهو اسم الدالة.**

**Parameters-:-وهي المتغيرات أو القيم التي يتم تمريرها إلى الدالة التي قد تكون خالية أحيانا.**

**Statements-:-يمثل جسم الدالة.**

**الثالث:- جزء الاستدعاءfunction calling :-**

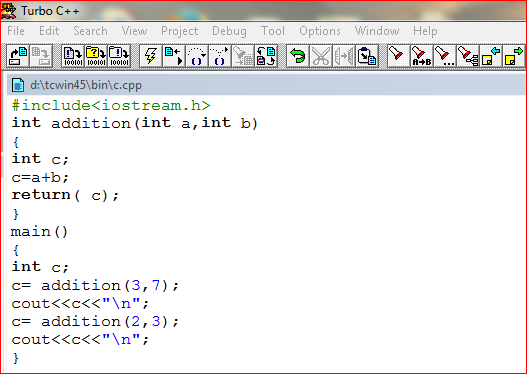
**يتم تنفيذ إي دالة بمجرد استدعائها حيث يؤدي الاستدعاء إلى انتقال التنفيذ إلى بداية الدالة المُستدعاة وإكمال تنفيذها ثم يعود إلى دالة الـmain() وتنفيذ العبارة التي تلي عبارة الاستدعاء لان جزء الاستدعاء يكون ضمن دالة الـmain() كما يمكن تمرير المتغيرات إلى الدالة عند استدعائها.**

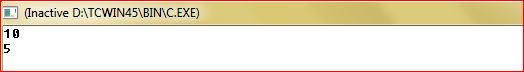
**بعض الملاحظات :**

* **قد يحتوي إي برنامج على دالة أو اثنين أو أكثر وقد لا يحتوي على دالة ماعدا دالة الـmain() إذ لا يمكن لأي برنامج إن لا يتضمنها لأن تنفيذ إي برنامج يعتمد أو يبدأ بها.**
* **إذا كانت القيم غير معرفة في البرنامج فيجب استخدام إيعاز cin>> في جزء main() فقط وذلك من أجل إدخال قيم المتغيرات .**
* **إذا كانت الدالة المستخدمة من نوع DataTypy (int ) فأن هذه الدالة ترجع قيم عن طريق استخدام return .**

**Q1 : Write a program to collect two numbers ,using function.**

**Solution :**

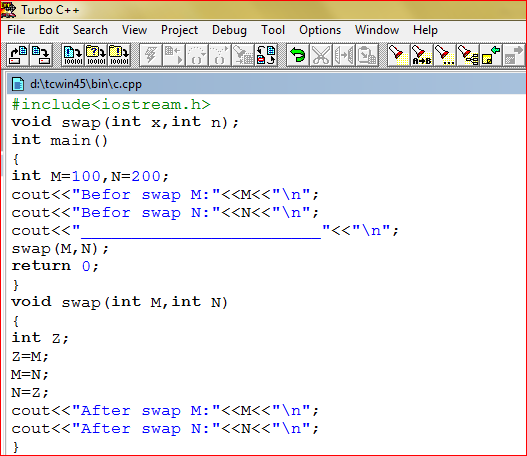
****

****

**ليس بالضرورة إن تكون أسماء التغيرات هي نفسها في جزء الإعلان وجزء التعريف وجزء الاستدعاء فقد تختلف من جزء لأخر لكن كونها مُتشابهَ يكون لك مفيد لمن يَقرَءون البرنامج إذ يسهل عليهم فهمه.**

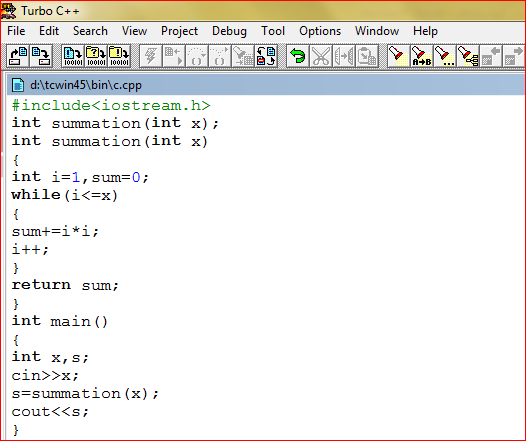
**Q 2: Write a program to swapping between two numbers .using function.**

**Solution :**

****

****

**Q3: write a program to find the following series.**

****

**أنواع الدوال :-**

**أولا:- pass by value أي إن الدالة تستقبل cin>>.**

**ترسل فقط قيمة المتغير إلى الدالة أي إذا تغيرت قيمة المتغير داخل الدالة لا تتغير القيمة الأصلية داخل البرنامج لأننا أرسلنا فقط القيمة إلى الدالة لمعالجتها إي (لا تعاد أكثر من قيمة واحدة إلى البرنامج).**

**ثانياً:- pass by reference أي إن الدالة تستقبل وتخرج قيمة cin>>&cout<<.**

**الإرسال بالمرجع, نرسل موقع المتغير إلى الدالة أي أن الدالة تستقبل المتغير نفسه المرسل بنفس الاسم أو باسم أخر إي إذا تغيرت قيمة المتغير داخل الدالة تتغير قيمته الأصلية داخل البرنامج لأننا أرسلنا موقعه إلى الدالة والتغيير يكون في محتوى الموقع .**

**Data\_ Type Function\_ Name (int & X, int &Y)**

**ثالثاً:- out parameters أي إن الدالة تخرج قيمة.**

**إذا كانت الـDataType الذي يسبق أسم الدالة هو(void)هذا يدل على أن الدالة لا ترجع قيمة أما أذا كان أي نوع أخر هذا يعني أن الدالة ترجع قيمة .**

**أذا كانت الـ Data Typeالموجودة داخل أقواس الدالة هي (void)أو كانت فارغة هذا يعني أن الدالة لا تستقبل**

**أما أذا اِحتوت نوع أخر يدل هذا على أن الدالة تستقبل.**

**طرق الاِستدعاء:-**

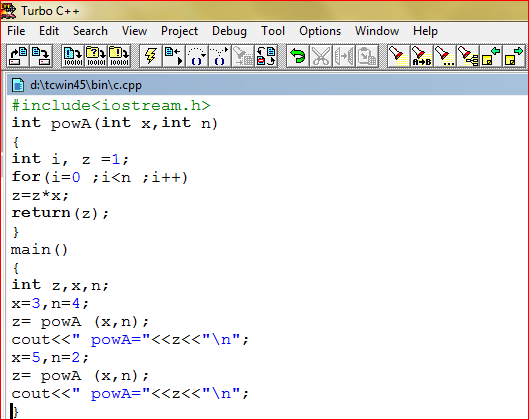
**أولا:- اِستدعاء الدوال بواسطة القيم .**

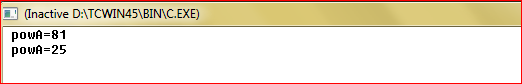
**حيث تقوم الدالة الرئيسية بإرسال قيم المتغيرات إلى الدالة التي تم اِستدعائها وتستقبل الدالة المُستدعاة هذه القيم و إرجاع قيمة معينة إلى دالة الـmain() .**

**ثانياً:- الاِستدعاء الذاتي**

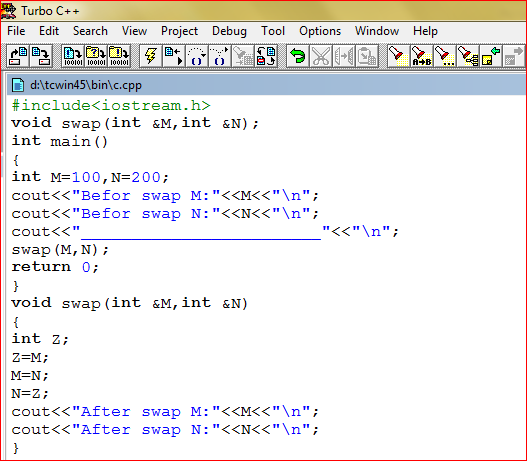
**أي أن الدالة تقوم باِستدعاء نفسها وهذه الخاصية مفيدة في حل المسائل التي تحتاج إلى اِستدعاء متكرر.**

**Q4: Write a program to calculate square number ,using function**

****

****

**Q 5: Write a program to swapping between two numbers .using function.**





**المتغيرات Argument**

* **Formal argument المتغيرات الرسمية .**

**المتغيرات الرسمية أو الوهمية وهي متغيرات تقع في جزء تعريف الدالة أو إعلان الدالة فقط .**

* **Actual argument المتغيرات الحقيقية .**

**وتقع هذه المتغيرات في جزء استدعاء الدالة**

* **Local and Global variable المتغيرات المحلية والعالمية**
* **Local variable المتغيرات المحلية**

**وهي متغيرات تعرف داخل الدالة .**

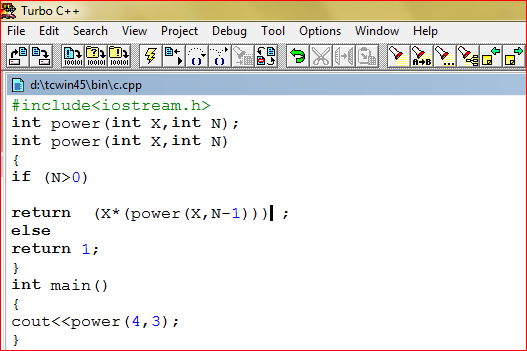
* **Global variable المتغيرات العالمية**

**هي المتغيرات التي تعرف خارج جزء main() في البرنامج حيث تكون معرفة في البرنامج ومتاحة الاستخدام في البرنامج .**

**Recursive function دالة الاستدعاء الذاتي**

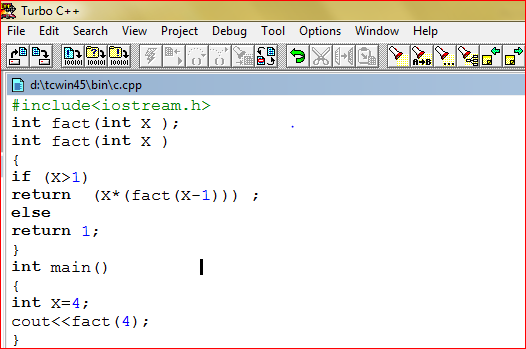
**هي دالة تستدعي نفسها أكثر من مرة مباشرة أو غير مباشرة حيث يعتبر هذا النوع من الدوال مهم جدا في هياكل البيانات .**

**Q6: Write a program to calculate square number ,using Recursive function**

****

****

**Q7: Write a program to calculate factorial number ,using Recursive function**

****

****

**السلاسل الرمزية (string &char)**

**إن الحاجة الماسة للتعامل مع السلاسل الرمزية , أجبرت مستخدمي c++ إن يوفروا طريقة يمكن من خلاله التعامل مع السلاسل الرمزية , والطريقة هي استخدام النوع الرمزي (char) من خلال الإعلان عن مصفوفة جميع عناصرها من النوع الرمزي زمن خلال هذه , الطريقة في التعامل مع السلاسل الرمزية , نستطيع إن نعرف السلاسل الرمزية على أنها مجموعة من الرموز مرتبة بشكل متسلسل على إن تنتهي ب Null (10) .**

**الإعلان عن السلاسل الرمزية**

**إحدى الطرق الممكن اتخاذها في الإعلان من السلاسل الرمزية هي المصفوفات استخدام المصفوفات في الإعلان عن السلاسل الرمزية .**

**يتم الإعلان عن السلاسل الرمزية باستخدام المصفوفات من خلال الإعلان عن مصفوفة جميع عناصرها من النوع (char) بحيث كل عنصر من عناصرها يحتوي على حرف واحد فقط ويتم الإعلان كالآتي :-**

**Char name\_ array [size]**

**حيث إن name\_ array هو اسم المصفوفة التي نوعها (char) وعدد عناصرها size مثلا :**

**Char a[7]="Nasser";**

**أي إن المصفوفة مكونه من 7 عناصر 6 منها الحروف والعنصر الأخير هو Null وبما انه البيانات معروفة يمكن عدم ذكر حجم المصفوفة إي في حالة الإدخال المباشر فقط .**

**Char a[ ]="Nasser";**

* **بعض الملاحظات المهمة عند التعامل مع السلاسل الرمزية :-**

**1-لا يمكن إن يكون حجم المصفوفة اقل من الحروف المدخلة أي لو كانت لدينا المصفوفة التالية :-**

**Char a[4 ]="Nasser";**

**في هذه الحالة التعبير خاطئ لأنه سوف يتم حجز مواقع فقط في الذاكرة وان السلسلة مكونة من 6 حروف مع Null تكون 7 .**

**2-في حالة الإدخال المباشر للبيانات فانه يجوز عدم ذكر حجم المصفوفة كما في المثال التالي :-**

**Char a[ ]="Nasser";**

**3-عند التعامل مع السلاسل الرمزية هناك عدة طرق من المصفوفات والمؤشرات ولا يجوز استخدامها معا أي إما الإعلان باستخدام المصفوفات أو الإعلان باستخدام المؤشرات :-**

**X\*="Ali" or X[ ]="Ali"**

**4- عند إدخال السلاسل الرمزية بطريقة حرف حرف فانه لابد من ذكر حجم المصفوفة :-**

**Char a[ 4]= { 'A' , 'l' , 'i' , '\0' };**

**5- لو أردنا الإعلان عن سلسلة رمزية بدون استخدام المصفوفات فان ذلك لا يتم مع استخدام Char وإنما نستخدم string :-**

**Char x= "Ali" ; False , string x="Ali"; True**

**6- إن الرمز Char يستعمل لإدخال حرف واحد فقط مصحوب بعلامة اقتباس مفردة إما string يستعمل لمجموعة من الحروف مصحوب بعلامة اقتباس مزدوجة مع انه يمكن استخدام Char لإدخال سلسلة من الحروف باستعمال المصفوفات كما ذكرت سابقا**

**Char x="M"; string x="Mohammed";**

**Char x[ ]="Ali";**

**7-عند استعمال Cin لإدخال بيانات السلاسل يتم ذلك عن طريق ذكر اسم المصفوفة فقط التي سيتم فيها تخزين حروف السلسلة المدخلة دون ذكر حجمها هنا تأتي مسؤولية المبرمج في أقلة المصفوفة التي سيتم تعريفها لتخزين السلسلة يجب إن تكون كبيرة لما يكفي لتخزين السلسلة التي يدخلها المستخدم عن طريق لوحة المفاتيح ويجب إن نذكر إن Cin حالما يجد فراغ يتوقف عن قراءة المدخل ويقوم بتخزين السلسلة المدخلة في المصفوفة المعلن عنها لتخزينها**

**Char x[4 ] ;**

**Cin>>x;**

**8-وأيضا عند استخدام cout لا نذكر حجم المصفوفة حيث نقوم بطباعة حروف السلسلة حتى تصل Null الذي يحدد نهاية السلسلة .**

**Char x[4 ] ;**

**Cin>>x;**

**Cout<<x;**

**الدوال المستخدمة مع Char & String**

**هناك عدة دوال تستخدم مع السلاسل الرمزية وان هذا الدوال تقع ضمن مكتبه خاصة وهي مكتبة <String.h>ومكتبة <cstring> لذالك لا بد من تضمين هذه المكتبات في إي برنامج يستخدم هذه الدوال فان الدوال هي :-**

1. **دالة strlen() :- لإيجاد عدد رموز السلسلة تستخدم بهذا ألصيغه strlen(x); .**
2. **دالة strcut():- لدمج سلسلتين ,ندمج x1,x2 بشرط إن x1 يستوعب حجم السلسلتين معا كما يلي strcut(x1,x2); .**
3. **دالة strcpy() :- لنسخ سلسلة إلى أخرى ,أي ننسخ محتويات x1إلى x2كما يلي strcpy(x1,x2) .**
4. **دالة strcmp() :- للمقارنة بين محتويات سلسلتين أي للمقارنة بين محتويات x1,x2 كما يلي strcmp(x1,x2) حيث تعد قيمة موجبة إذا كانت x1 اكبر وتعد قيمة سالبة إذا كان x2 اكبر وتعتبر صفر إذا**

**كانت القيمتين متساويتين .**

1. **دالة strlwr():- تحويل عناصر السلسلة من الحروف الكبيرة إلى الصغيرة .**
2. **دالة strupr():- لتحويل عناصر السلسلة من الحروف الصغيرة إلى الحروف الكبيرة.**
3. **Size of ():- عند التعامل مع البيانات وتخزينها في الذاكرة فإنها تخزن بصيغة bits حيث إذا كانت البيانات من نوع String فإنها تأخذ bits 4 بغض النضر عن طول السلسلة حيث إن كل ما موجود داخل علاقتي الاقتباس يخزن في bits 4 إما إذا كانت البيانات من نوع Char فان عدد bits يكون بعدد الحروف فإذا كان حرف واحد bits 1 وإذا كان مجموعة من الحروف معرفة بصيغة مصفوفة فان عدد bits يكون بعدد الأحرف +واحد الذي هو Null وحساب عدد bits في السلسلة يكون باستعمال دالة size of() فمثلا :-**

**Char s[ ] = {" bcdef"};**

**6+1(null)**

**Cout<<size of (s);**

**=7**

**Char x[ ]= { "a b c" };**

**Cout<<size of(x);**

**=4**

**بعض الملاحظات:-**

**الفرق بين strlen() و size of() حيث إن :-**

1. **size of() تعتمد على نوع data type للمتغير إما strlen() يعتمد على عدد الحروف المدخلة.**
2. **يجب الإشارة إلى انه إدخال البيانات بصيغة String فلا يجوز إن تحسب strlen() وذلك لأنها تعتبر (blook) ثابت كما ذكرنا سابقا إما إذا كانت char فان strlen() لها عدد الأحرف 1+**

**بعض العمليات التي تجري على البيانات التي تكون من نوع string:-**

**1-عملية Assign**

**S1=s2**

**إي نقل قيمة المتغير s2 إلى S1**

**2-عملية Appending**

**S1=s1+s2 or s1+=s2**

**يعني إضافة قيمة s2 إلى s1 وبعبارة أخرى جمع قيمة s2 مع الصفر وإضافتها إلى s1**

1. **عملية Indexing**

**S1=s2[2,1]**

**إي لو كانت لدينا قيمة S2={"A B C D"};**

**فان عملية Indexing S1 هي يترك أول حرفين ويأخذ حرف واحد أي قيمة S1 تصبح c**

1. **عملية concatenate**

**أشبه بدالة الجمع s1=s2+s3**

**جمع s2 مع s3 وإضافتها إلى s1**

**ملاحظة :-عند تعريف إي متغير بصيغة string والتي تعني البيانات تعتبر bloke غير قابلة للتجزئة عند طلب طباعة نصف العناصر فانه يتجاهل الشرط ويطبع الجملة بالكامل بدون تجزئة لأنها bloke أي لو كان لدينا المثال التالي :-**

**Char s1[ ] = " A B C D E F ";**

**For (int i=0; i<3; i++)**

**Cout<<s[ I] ;**

**في هذه الحالة يطبع أول ثلاث حروف فقط لان البيانات من نوع char**

**String s1 [ ] ="A B C D E F";**

**For (int i=0; i<3; i++)**

**Cout<<s[ i] ;**

**في هذه الحالة يتجاهل إل compiler شرط For ويطبع الجملة كاملة لان البيانات من نوع string**

**العمليات التي تجري على البيانات التي تكون من نوع char**

1. **عملية copy تكتب بصيغتين :-**

**الأولى : strcpy أي لو كانت لدينا قيمة x هي Char x = "A B C";**

**فان strcpy(z,x)**

**أي نقل قيمة x بالكامل إلى المتغير z والذي من نوع Char أيضا .**

**الثانية : strncpy تعني نقل n من الحروف إلى المتغير الذي يحمل القيمة إلى المتغير الثاني شرط إن يكون المتغير من نوع Char أي strncpy(z,x,1)**

**في هذه الحالة سوف ينتقل حرف واحد من قيمة x إلى z.**

1. **عملية cat أيضا تكون بصيغتين:-**

**الأولى : strcat() تقوم بنسخ كل محتويات المتغير إلى متغير ثاني كالأتي:-**

**Strcat(x,y)**

**تقوم بنسخ كل محتويات y إلى x**

**الثانية : strncat() تقوم بنسخ n من الحروف من المتغير إلى متغير أخر أي**

**Strncat(x,y,2)**

**أي تقوم بنسخ 2 من y إلى x**

1. **عملية compare :- تقوم بعملية المقارنة بين جملتين وتكو لها ثلاث حالات تكتب بالصيغة التالية strcmp() أي لو كانت**

**Char x=" A B C";**

**Char y=" A B C D";**

**Strcmp(x,y)**

**الحالات :-**

1. **إذا كان مصدر المقارنة اقل من المتغير الثاني فان ناتج المقارنة يكون 1- .**
2. **إذا كان مصدر المقارنة اكبر من المتغير الثاني فان ناتج المقارنة يكون 1+ .**
3. **إذا كان مصدر المقارنة يساوي المتغير الثاني فان ناتج المقارنة يكون صفر .**

**نقل البيانات التي تكون من نوع string or char**

**كثيرا ما نحتاج في c++ من عمليات نقل بين متغيرين أي من متغير معين إلى متغير أخر وان هذين المتغيرين إما يكونان من نوع string أو char أو أحداهما string والأخر char فان عملية نقل البيانات من متغير إلى أخر تتم بعدة شروط يجب إن تتوفر هذه الشروط لكي يكون النقل صحيح فإذا كانت البيانات من نوع string فان الطريقة الأنسب لنقل البيانات هي استخدام العمليات التي ذكرناها سابقا كالأتي :-**

**1) Assign**

**s1 = s2**

**مرسل مستقبل**

**String string**

**إذا كان مرسل البيانات من نوع String ومستقبل البيانات من نوع String يجوز ننقل البيانات بعملية Assign**

**-s1 = s2**

**String Char**

**إذا كان مرسل البيانات من نوع Char والمستقبل من نوع String يجوز نقل البيانات بعملية Assign.**

**-s1 = s2**

**Char string**

**إذا كان مرسل البيانات من نوع string والمستقبل من نوع Char لا يجوز نقل البيانات .**

**-s1 = s2**

**Char Char**

**إذا كان مرسل البيانات من نوع Char والمستقبل من نوع Char لا يجوز نقل البيانات .**

**2)Appending:-**

**s1 += s2**

**String string**

**إذا كان مرسل البيانات String والمستقبل من نوع String فانه يجوز نقل البيانات بعملية Appending**

**s1 += s2**

**String char**

**إذا كان مرسل البيانات char والمستقبل من نوع String فانه يجوز نقل البيانات بعملية Appending**

**s1 += s2**

**Char String**

**إذا كان مرسل البيانات String والمستقبل من نوع char فانه لا يجوز نقل البيانات بعملية Appending**

**s1 += s2**

**char char**

**إذا كان مرسل البيانات char والمستقبل من نوع char فانه لايجوز نقل البيانات بعملية Appending**

**3)Indexing**

**s1 = s2[ 3]**

**String String**

**إذا كان مرسل البيانات String والمستقبل من نوع String فانه يجوز نقل البيانات بعملية Indexing**

**s1 = s2[ 3]**

**String char**

**إذا كان مرسل البيانات String والمستقبل من نوع char فانه يجوز نقل البيانات بعملية Indexing**

**s1 = s2[ 3]**

**char String**

**إذا كان مرسل البيانات char والمستقبل من نوع String فانه لا يجوز نقل البيانات بعملية Indexing**

**s1 = s2[ 3]**

**char char**

**إذا كان مرسل البيانات char والمستقبل من نوع char فانه لا يجوز نقل البيانات بعملية Indexing**

**4)concatenate**

**s1 = s2 + s3**

**String string string**

**إذا كان نوع البيانات المراد جمعها من نوع string والناتج string فانه يمكن إجراء عملية concatenate**

**-s1 = s2 + s3**

**String char string**

**إذا كان نوع البيانات المراد جمعها من نوع string والأخر من نوع char والناتج string فانه يمكن إجراء عملية concatenate**

**-s1 = s2 + s3**

**String char char**

**إذا كان نوع البيانات المراد جمعها من نوع char والناتج string فانه لا يمكن إجراء عملية concatenate**

**s1 = s2 + s3**

**char char char**

**إذا كان نوع البيانات المراد جمعها من نوع char والناتج char فانه لا يمكن إجراء عملية concatenate**

**s1 = s2 + s3**

**char string char**

**إذا كان نوع البيانات المراد جمعها من نوع char والأخر من نوع string والناتج char فانه لا يمكن إجراء عملية concatenate**

**نلاحظ من العمليات السابقة ونوع البيانات انه إذا مستقبل البيانات من نوع char فانه لا يمكن إجراء العمليات السابقة في نقل البيانات في أي حالة من الحالات سواء أكانت مرسل البيانات من نوع char أو string ما عدا حالة واحدة يكون فيها مستقبل البيانات string والمرسلات char فانه أيضا لا يجوز إجراء العملية**

**s1 = s2 + s3**

**string char char**

**ولكن نعالج هذه المشكلة وهي إذا كان مستقبل البيانات من نوع char يتم ذلك باستخدام الدوال التالية (cmp(),cat(),cpy())**

**strcpy(s3,s1)**

**Char string**

**إذا كان مرسل البيانات من نوع string والمستقبل Char فانه لا يجوز نقل البيانات في دالة strcpy**

**strcpy(s3,s1)**

**Char char**

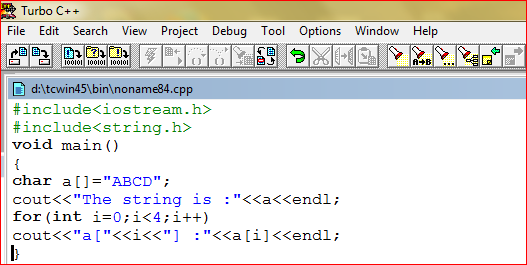
**إذا كان مرسل البيانات من نوع Char ومستقبل البيانات من نوع Char فانه يجوز نقل البيانات بدالة strcpy**

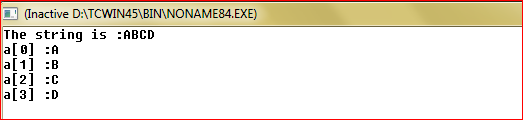
**- strcpy(s3,s1)**

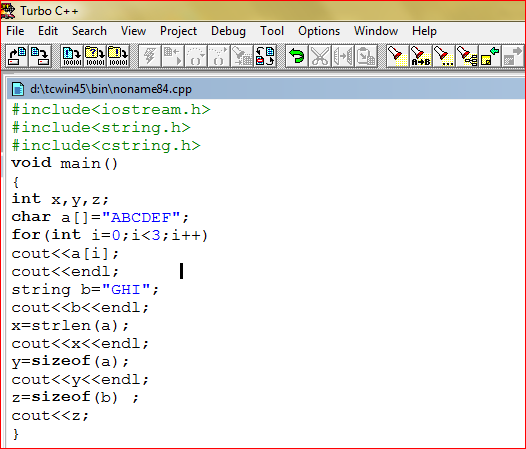
**String string**

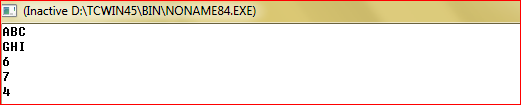
**إذا كان مرسل البيانات من نوع String ومستقبل البيانات من نوع String فانه نفس الحالة إذا كانت الدالة المستخدمة cat ,cmp**

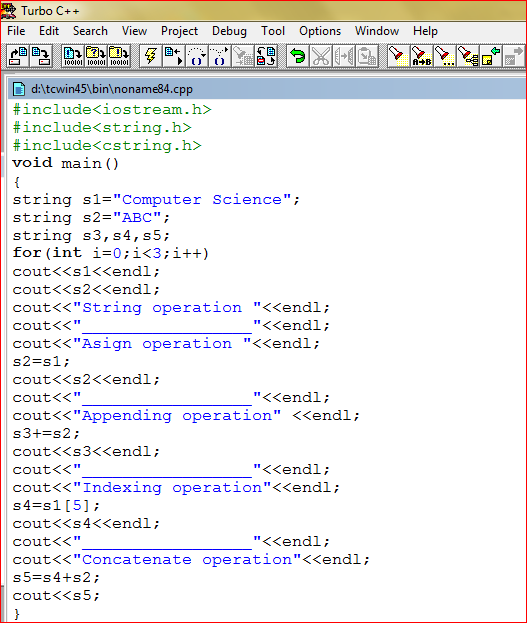
**Q1: Write a program to print string ,then print it character by character .**

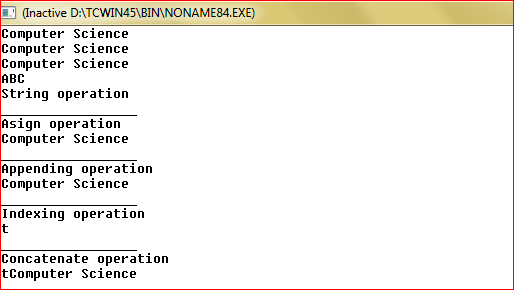
****

****

****

****

****

****

**توضيح للسؤال :**

**في العملية الأولى s2=s1 هنا يتم نقل بيانات s1 إلى s2 و تهمل بيانات s2 .**

**في العملية الثانية s3+=s2 يجمع بيانات s2 مع 0 ويضيفها إلى s3 .**

**في العملية الثالثة s4=s1[5] يترك خمس أحرف من s1 وينقل حرف واحد إلى s4 .**

**في العملية s5=s4+s2 يجمع بيانات s2 مع s4 .**

**التركيب structures.**

**تأتي أهمية التراكيب لتعريف متغيرات بطرق مختلفة فقد نحتاج إلى تعرف متغير له أكثر من نوع من البيانات , أسلوب تعدد البيانات لمتغير واحد نعبر عنه في المصفوفات لكن يجب أن تكون البيانات من نوع واحد ولتجاوز هذه المشكلة نستخدم التراكيب .**

**هيكلية تعريف التراكيب .**

**Struct Name**

**{**

**Different data type ;**

**};**

**Main()**

**{**

**Name struct object;**

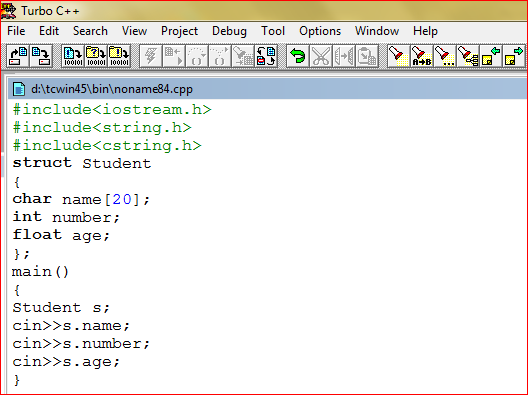
**}**

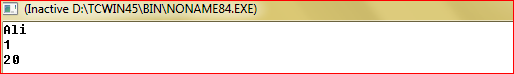
**Struct : كلمة محجوزة للدلالة على استخدام التراكيب .**

**Name : اسم التركيب ويفضل أن يبدأ بحرف كبير .**

**Different data type ; : أنواع المتغيرات التي ستعرف في البرنامج والتي تكون مختلفة في data\_ type .**

**object; : هو الذي يسمح للدخول للـ struct والتعرف على المتغيرات الموجودة داخله باعتبار الـ struct هو خارج main program وأي دالة أو متغير خارج( main( يحتاج تعريف داخل main() عن طريق ذكر أسم التركيب ومن ثم الـ objrct الذي يسمح بالدخول الى struct والتعرف على المتغيرات الموجودة داخلة .**

****

****

**توضيح السؤال:**

**في هذا السؤال يطلب إدخال بيانات طالب اسم رقم وعمر وبما أن البيانات مختلفة في data type استخدمنا struct , في جزء main() عرفنا object وهو sيسمح بالدخول إلى struct والتعرف ع المتغيرات , أن استخدام المتغير في جزء main() يجب أن يكون مصحوب بــ object قبل المتغير .**

**s.name**

**s. number**

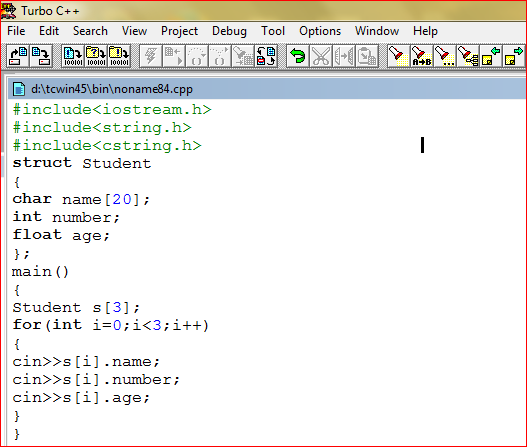
**s. age**

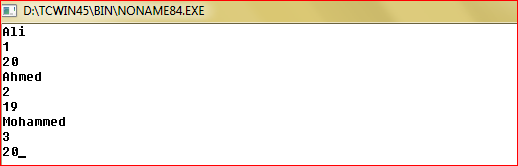
**هذه المعلومات لطالب واحد أما إذا طلب في السؤال لأكثر من طالب ؟**

**في هذه الحالة أما نستخدم أكثر من object (s1,s2,s3….) .**

**أو طريقة ثانية وهي أكثر استخداماً وهي كتابة الـ object بشكل مصفوفة s[3]**

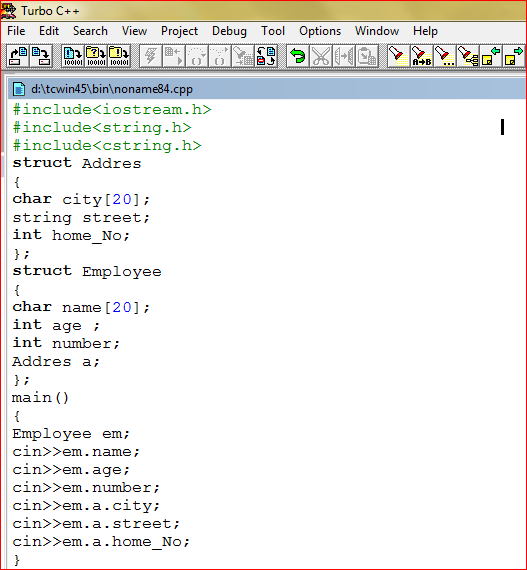
**كما في المثال الأتي.**

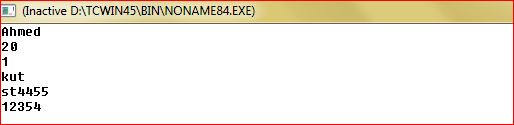
****

****

**التراكيب المتداخلة nested structure .**

**هي طريقة وضع تركيب داخل تركيب الفائدة منه هي مثلا لو لدينا متغير يحتوي على أكثر من معلومة أي لو طلب منا العنوان بالكامل فأننا نحتاج إلى struct خاص بالعنوان نضع فيه متغيرات العنوان كالمدينة ,اسم الشارع ,رقم الدار وغيرها . لنأخذ المثال الأتي .**

****

****