مدونة المناهج السعودية

https://eduschool40.blog

الموقع التعليمي لجميع المراحل الدراسية

في المملكة العربية السعودية

المملكة العربية السعودية
جامعة الملك سعود
كلية المعلمين بالرياض

مقدمة إلى الخوارزميات و البرمجة

إعداد الدكتور : عبد غازي

المملكة العربية السعودية

جامعة الملك سعود

كلية المعلمين بالرياض

بسم الله الرحمن الرحيم

مقدمة إلى الخوارزميات والبرمجة

إعداد الدكتور:عميد غازي

مفردات مقدمة إلى الخوارزميات والبرمجة

1.الخوارزميات:

- مقدمة عن مفهوم الخوارزميات .
- تعريف الخوارزمية.
- أنواع الخوارزميات.
- طرق التعبير عن الخوارزمية.
- برامج وتمارين عامة.

2. البرمجة بلغة C++::

- أنواع لغات البرمجة .
- التعرف على تاريخ لغة C++.
- التعرف على البرمجة المهيكلة Structured Programming.
- كيفية كتابة برنامج بلغة c++.
- قواعد تسمية الأسماء التعريفية للبرامج عند حفظها.
- الكلمات المحجوزة.
- أمثلة وتمارين بسيطة.
- التصريح عن المتحولات .
- أنواع المتحولات البسيطة (الصحيحة – الحقيقية – المحرفية – البوليانية-...)
- العمليات الحسابية وعمليات المساواة والمقارنة.
- عمليات الاسناد.
- أمثلة وتمارين

3. بنى التحكم Control Structures:

- بنية الاختيار if.
- بنية الاختيار if/else.
- بنية الاختيار المتعددة.
- البنى التكرارية :
  - البنية While .
  - البنية do/while .
  - البنية for.
- بنية الاختيار المتعدد switch.
- برامج وتمارين عامة .

المراجع العربية:

1. مفاهيم و أسرار لغة البرمجة c++ / أ.د علاء حسين الحمامي 2008/
2. كيف تبرمج بلغة ال++ c / ترجمة الدكتور صلاح الدوه جي 2002/
3. c++ الدليل الكامل / منشورات الدار العربية للعلوم2002 /
4. المدخل إلى علم الحاسوب / د.عبد الرحمن العبد،د بشرى سماقية 2003/

المواقع:

www.arabteam200.com

www.cprogramming.com

www.c++LanguageNotes.com

www.odffactorv.com

i. الخوارزمية (algorithm)

أصل كلمة خوارزمية :

إن كلمة خوارزمية مشتقة من إسم العالم العربي الجليل محمد بن موسى الخوارزمي الذي عاش في بغداد من سنة 780 الى 847م في عصر الخليفة المأمون , وقد برع هذا العالم في الرياضيات والفلك , وترك بصمات في التراث الحضاري العالمي , فقد وضع الخوارزمي مبادئ علم الجبر وألف كتاب "الجبر و المقابلة" وأعطى الجبر اسمه حتى اصبحت كلمة الجبر موجودة في جميع اللغات تقريباً

وفي تلك الأونة انطلق اسم الخوارزميات Algorismsعلى جداول الضرب والقسمه والحساب العشري , وظل هذا الاسم متداولاً في أوروبا مدة قرون حتى تطور مؤخراً ليحمل مدلولاً جديداً مرتبطاً بالبرمجة .

1. مقدمة :

إن أهم مرحلة في حل مسألة ما بإستخدام الحاسوب هي المرحلة المتعلقة بإيجاد خطة الحل , يجب أن تكون هذه الخطة قابلة للتنفيذ من قبل الآلة , وقابلة للتوصيف على وجهٍ لا يدعو الى اللبس أو التأويل , يطلق اسم الخوارزمية على هذه الخطة .

2. تعرف الخوارزمية:

مجموعة الخطوات المتسلسلة والمحدودة التي تؤدي إلى حل مسألة معينة والوصول إلى نتائج محددة اعتباراً من معطيات ابتدائية.

3. أنواع الخوارزميات:

1) **خوارزميات حسابية**: تهتم بالمسائل الرياضية . (حل معادلة من الدرجة الأولى).

2) **خوارزميات غير حسابية**: لا تهتم بالمسائل الرياضية ولكنها تحتاج إلى حل منطقي.

(طريقة التدقيق الإملائي لنص ما،اتخاذ قراربالذهاب إلى مكان ماوتحديد الطريق الأمثل للوصول إليه).

سنهتم في هذا الفصل بالخوارزميات الحسابية فقط.

4. طرق التعبير عن الخوارزمية :

1) **الطريقة الكلامية :** كتابة الخوارزميات على شكل خطوات باستخدام اللغة المتداولة كاللغة العربية أو الإنكليزية.

2) **الطريقة الرمزية :** كتابة الخوارزميات باستخدام الرموز.

3) **الطريقة التدفقية :** كتابة الخوارزميات باستخدام المخططات البيانية (المخططات التدفقية).

5. مثال توضيحي :

أكتب الخوارزمية التي تعطي نتيجة حل التعبير الرياضي الآتي باستخدام اللغة المتداولة (الطريقة الكلامية):

$$Y=(x^2+7)/x(x+2)$$

علماً بأن x معلومة .

الحل:

يمكن التعبير عن الخوارزمية باللغة المتداولة(العربية) على الشكل الآتي:

الخطوة الأولى : أقرأ(أدخل) قيمة المتحول x .

الخطوة الثانية: احسب المقام : a=x(x+2)

الخطوة الثالثة: إذا كان المقام مساوياً للصفر اطبع " المسألة ليس لها حل" .

الخطوة الرابعة: وإلا احسب البسط : $b=(x^2+7)$

الخطوة الخامسة: احسب قيمة y : y =b /a .

الخطوة السادسة: اطبع (أكتب) قيمة y.

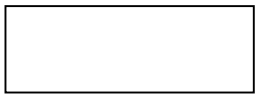
الخطوة السابعة: توقف .

6. المخطط التدفقي (الهندسي أو الكابي):

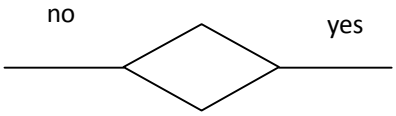
لبناء المخطط التدفقي نستخدم مجموعة من الأشكال الهندسية لتسهيل هذا المخطط :

1. عمليات الإدخال والإخراج نستخدم الشكل :

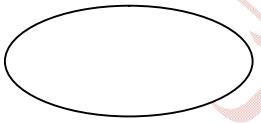
2. عملية المعالجة نستخدم الشكل :



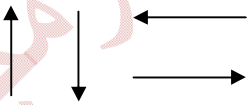
3. عملية الشرط(القرار) نستخدم الشكل :



4. لبداية ونهاية الخوارزمية نستخدم الشكل :



5. لمعرفة اتجاه الخوارزمية نستخدم الشكل :



6. نقطة توصيل وربط :

تمارين :

تمرين1: اكتب الخوارزمية الكلامية والرمزية والمخطط التدفقي لإيجاد مساحة ومحيط المستطيل ؟.

الحل :

الخوارزمية الكلامية :

1- المدخلات : أدخل(اقرأ): الطول والعرض .

2 - المعالجة : المساحة(s) = الطول x العرض

المحيط (m) (الطول+ العرض) x 2

3- المخرجات : أطبع قيمة المساحة والمحيط

الخوارزمية الرمزية :

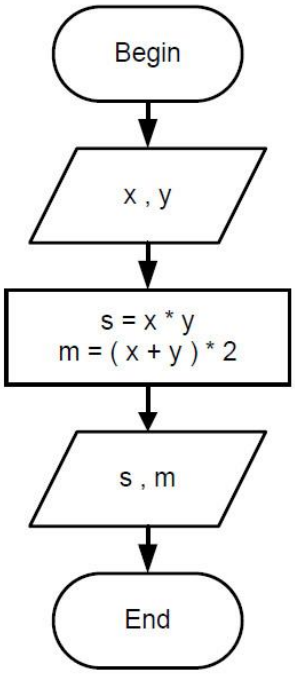
1- المدخلات: أدخل(اقرأ): y, x

2- المعالجة : s= y*x

m = (y+x) * 2

3- المخرجات : أطبع قيمة m , s

المخطط التدفقي :



تمرين2: على نمط المثال السابق اكتب الخوارزمية الكلامية و الرمزية والمخطط التدفقي لإيجاد مساحة ومحيط الدائرة ؟

الحل :

الخوارزمية الكلامية :

1. المدخلات: أدخل(اقرأ): نصف القطر (r)
2. المعالجة : المساحة(s) = π x نصف القطر للتربيع

   المحيط (m) =2 x نصف القطر p x
3. المخرجات: أطبع قيمة المساحة والمحيط لدائرة

الخوارزمية الرمزية :

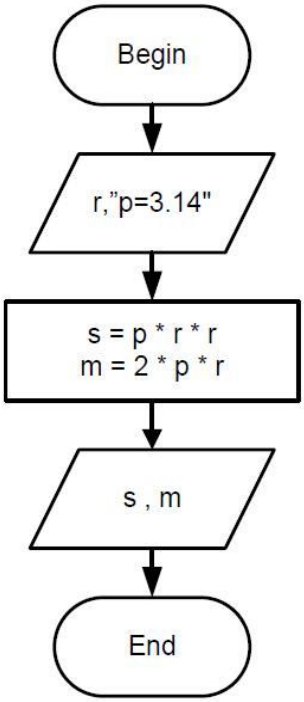
1- المدخلات : أدخل(اقرأ): r

2- المعالجة : s= p* r * r

m= p* r *2

3- المخرجات : أطبع قيمة s,m

المخطط التدفقي :



تمرين3: **اكتب الخوارزمية الرمزية والمخطط التدفقي لإدخال x (عدد) وإيجاد قيمة** $y=(x-2)/x$

الحل:

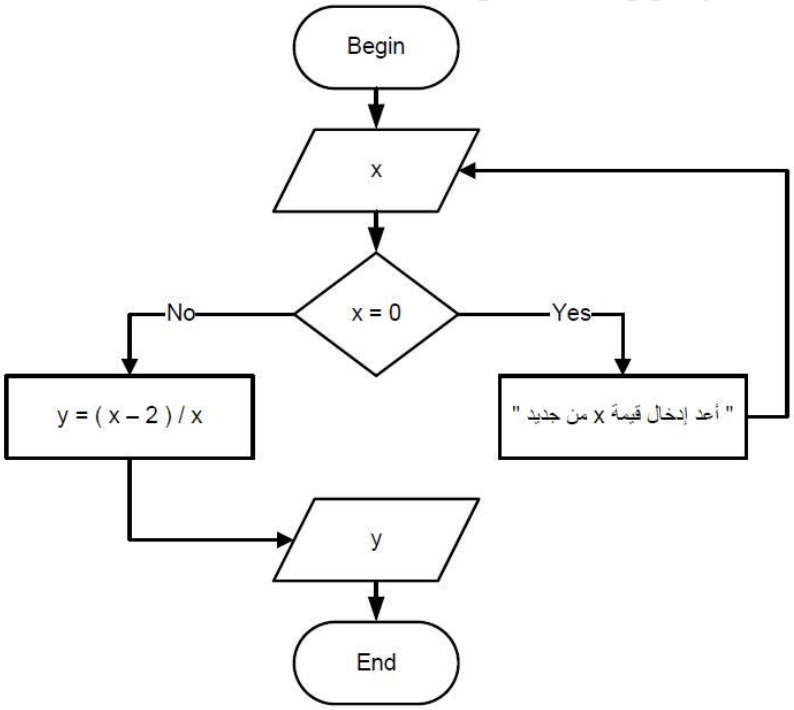
الخوارزمية الرمزية : 1) المدخلات : أدخل(اقرأ): x

2) المعالجة : إذا كانت (x=0) عندئذ " أعد ادخال قيمة x من جديد لانه لايمكن القسمة على صفر"

وإلا فاحسب : $y=(x-2)/x$

3) المخرجات : أطبع قيمة y

المخطط التدفقي :



تمرين4: **اكتب الخوارزمية الرمزية والمخطط التدفقي لايجاد y=x/(x-3)**

الحل:

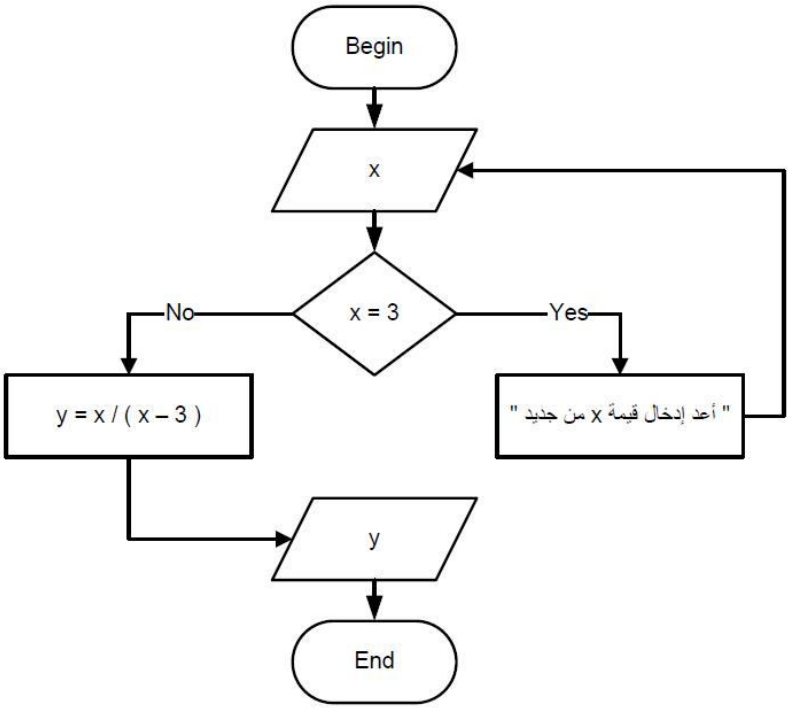
الخوارزمية الرمزية : 1) المدخلات : أدخل(اقرأ): x

2) المعالجة : إذا كانت (x=3) عند ئذ " اعد ادخال قيمة x "

وإلا أحسب y=x/(x-3)

3) المخرجات : أطبع قيمة y

المخطط التدفقي :



تمرين 5: اكتب الخوارزمية الرمزية والمخطط التدفقي لحل المعادلة :**a X + b = 0**

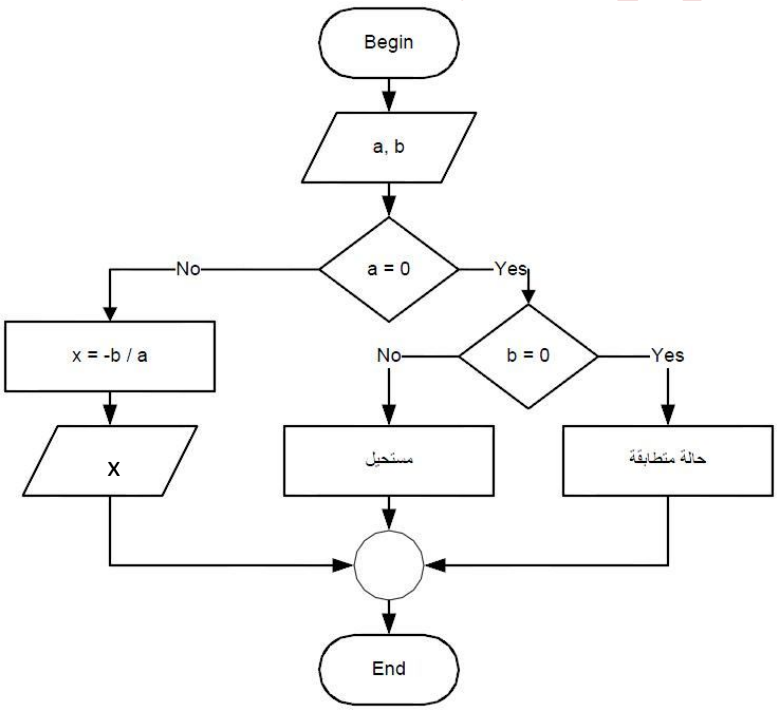
مناقشا جميع الحالات الممكنة لـ a,b

الحل :

الخوارزمية الرمزية :

1) المدخلات : أدخل(اقرأ): a , b .
2) المعالجة و المخرجات : إذا كان (a=0 , b< >0) أطبع : "مستحيل الحل".
وإلا إذا كان (a=0 , b=0) أطبع : " حالة متطابقة "
وإلا (a< >0) نجد : x=-b/a
أطبع قيمة x

المخطط التدفقي :



تمرين 6: **اكتب الخوارزمية الرمزية والمخطط التدفقي (الانسيابي) لإيجاد قيمة y المعطاة بالشكل التالي :**

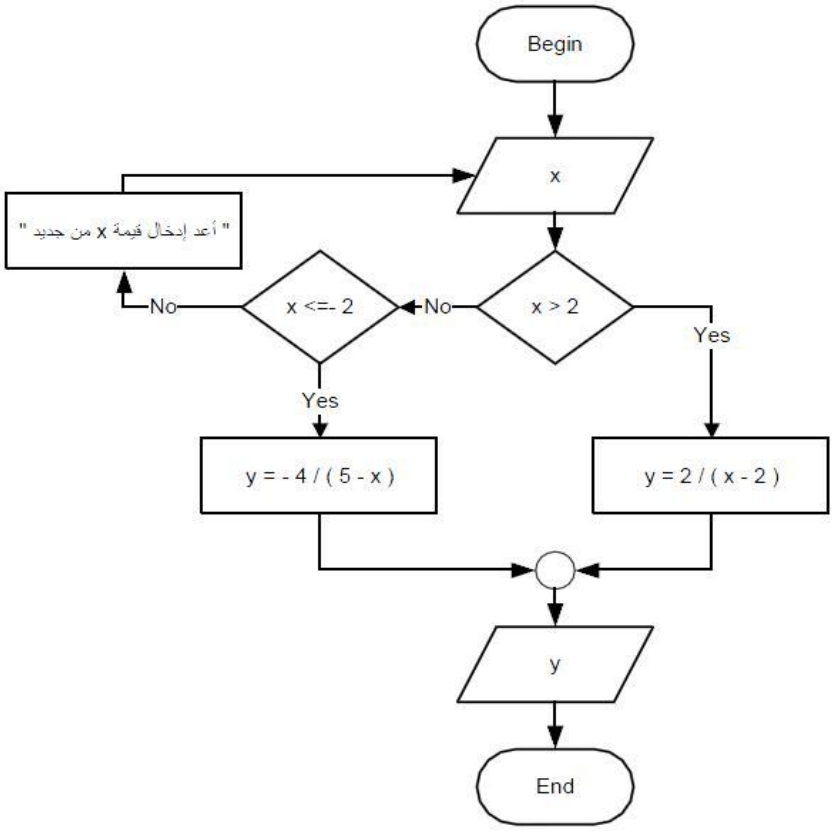
$$Y=\begin{cases} 2/(x-2) & x>2 \\ -4/(5-x) & x<=-2 \end{cases}$$

الحل :

الخوارزمية الرمزية :

1- **المدخلات** : أدخل(اقرأ): x
2- **المعالجة** : إذا كانت (x>2) عندئذ y=2/(x-2)
وإلا إذا كانت (x<=-2) عندئذ y=-4/ (5-x)
وإلا " اعد ادخال قيمة x "
3- **المخرجات** : أطبع قيمة y

المخطط الانسيابي (التدفقي , الصندوقي) :



تمرين 7: اكتب الخوارزمية والرمزية والمخطط التدفقي لحل معادلة الدرجة الثانية
$aX^2+bX+c=0$

الحل :

الخوارزمية الرمزية :

1. أدخل(اقرأ): a,b,c
2. إذا كان (a=0) نفذ :

تصبح المعادلة معادلة من الدرجة الأولى : $bX+c=0$

i) إذا كان: (b = 0)

إذا كان: (C =0) أطبع " حالة متطابقة"

إذا كان: (C < >0) أطبع " حالة مستحيلة"

ii) إذا كان: (b <> 0)

أطبع قيمة X= -c/b

3. إذا كان (a<> 0)

حساب D دالتا : $D=b^2 -4*a*c$

i) إذا كان: (D = 0)

أطبع : " للمعادلة جذران متماثلان "

وأحسب: $X_1 = X_2=-b/2*a$

ii) إذا كان: (D < 0)

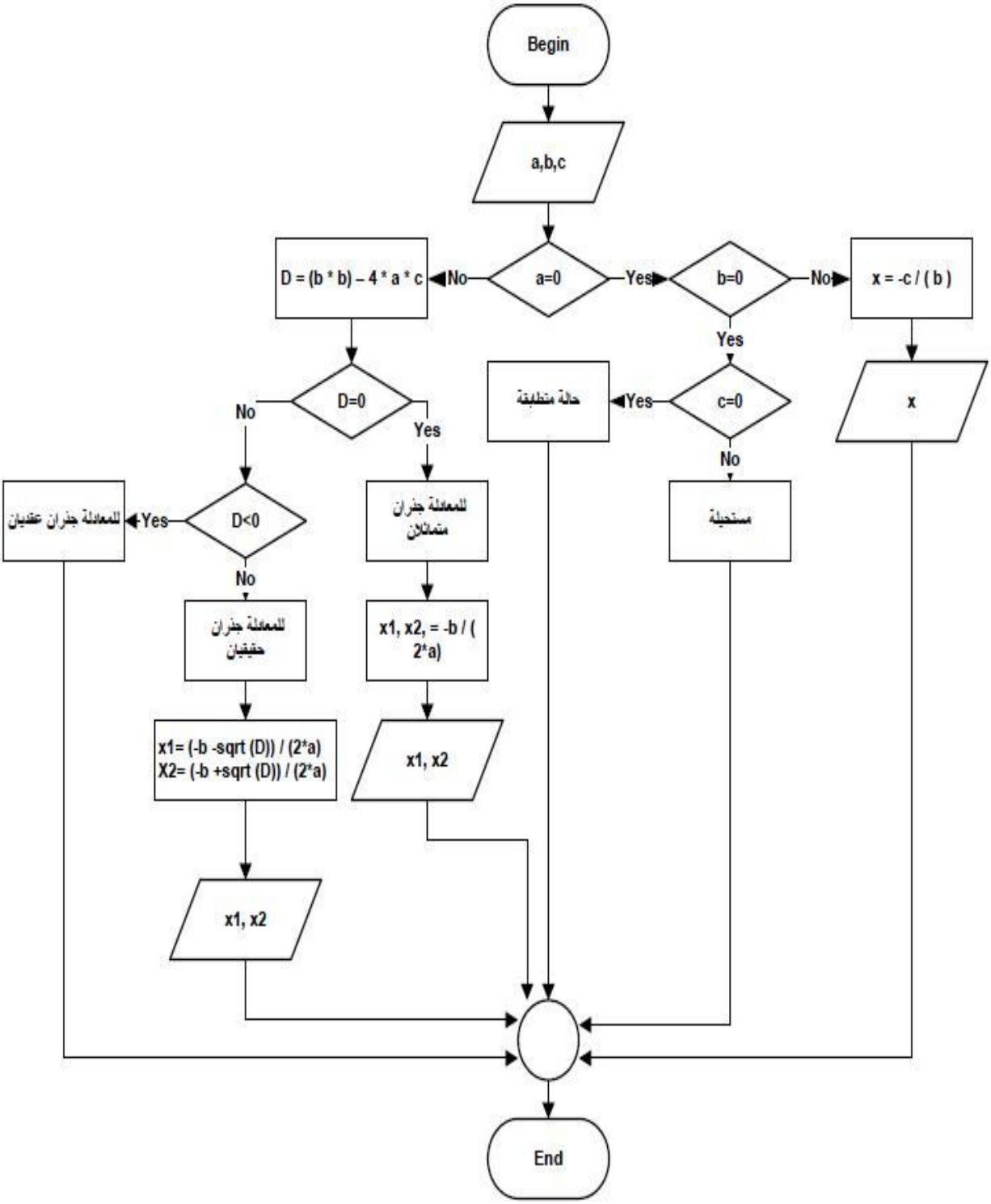
أطبع : " للمعادلة جذران عقديان "

iii) إذا كان: (D > 0)

أطبع : " للمعادلة جذران حقيقيان "

أطبع قيمة: $X_2=(-b+sqrt(D))/(2*a)$ $X_1=(-b-sqrt(D))/(2*a)$

المخطط التدفقي:



تمرين 8: **اكتب الخوارزمية الرمزية والمخطط التدفقي لإدخال عدد صحيح (x) موجب وطباعة إذا كان فرديا أم زوجيا ؟**

الحل :

الخوارزمية الرمزية :

1. **المدخلات:** أدخل(اقرأ): x

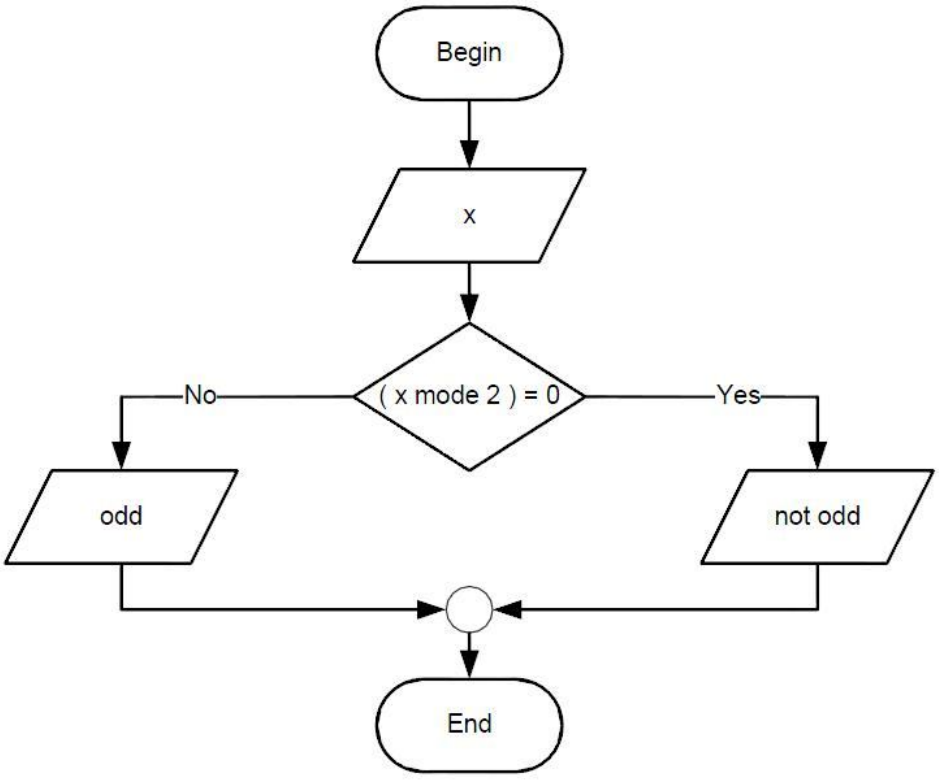
2. **المعالجة والمخرجات:** إذا كان باقي قسمة العدد على 2 يساوي صفر (x mode 2= 0) فإن :

أطبع " العدد زوجياً أو not odd "

وإلا فإن :

أطبع " العدد فردياً أو odd "

المخطط التدفقي :



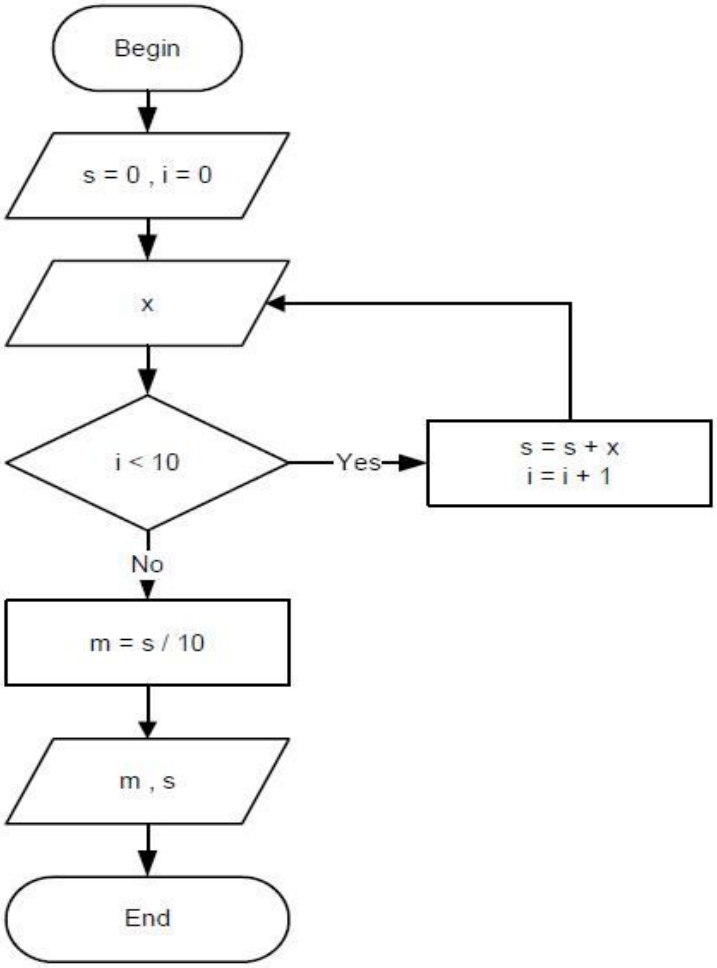
تمرين 9: اكتب الخوارزمية الرمزية والمخطط التدفقي لإدخال عشرة أعداد مختلفة وإيجاد المتوسط والمجموع ؟

الحل :

الخوارزمية الرمزية :

1. **المدخلات** : أدخل(اقرأ): x , S =0 , i=0
2. **المعالجة** : العداد (i=i+1); المجموع (S=S+x)
   إذا كان i<10 عندئذ " أعد إدخال x "
   وإلا i >=10 عندئذ " توقف عن الإدخال i " : m=S/10وأحسب
3. **المخرجات** : أطبع قيمة: المجموع (s) , المتوسط (m)

المخطط التدفقي :



تمرين10: اكتب الخوارزمية الرمزية والمخطط التدفقي لإدخال عشرة أعداد وطباعة الفردي منها فقط ؟

الحل :

الخوارزمية الرمزية :

1- **المدخلات** : أدخل(اقرأ): X, i=0

2- **المعالجة و المخرجات**: إذا كان (i<10)عندئذ

وإذا كان (X mode 2=0) عندئذ

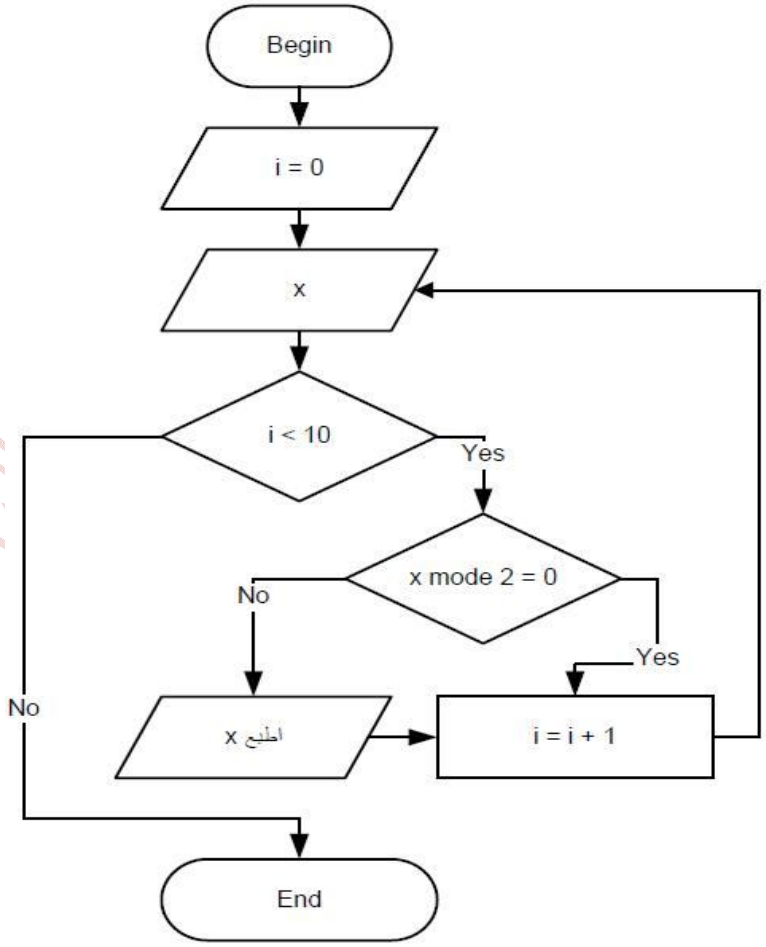
i=i+1 و" أعد إدخال x "

وإلا أطبع قيمة x الحالية ثم أدخل قيمة جديدة لـ x

i=i+1 وشغل العداد.

وإلا أخرج من البرنامج

المخطط التدفقي :



ii. البرمجة بلغة C++

* أنواع لغات البرمجة :

يمكن تقسيم اللغات المستخدمة في البرمجة إلى أربعة أنواع:

1- لغة الآلة ، 2- لغة المجمع ، 3- اللغات العالية المستوى ، 4- لغات البرمجة المرئية

1. لغة الآلة:

هي اللغة الوحيدة التي يستطيع الحاسب أن يفهمها مباشرة وهي معرفة من قبل البنية الصلبة للحاسب ، تتألف بشكل عام من سلاسل من الأعداد (مجموعات من الأصفار والواحدات) التي تعطي الأوامر للحاسب من أجل تنفيذ تعليماته الأولية كل تعليمة على حده.

2. لغة المجمّع :

هي لغة تستخدم مصطلحات قريبة من اللغة الإنكليزية للتعبير عن العمليات الأولية للحاسب، و يتم تحويل البرامج من لغة المجمع إلى لغة الآلة بواسطة مترجم خاص بها يسمى المجمّع Assembler.

3. لغات البرمجة العالية المستوى :

هي اللغات التي ظهرت مع ظهور المترجمات(Compiler : يقوم بترجمة لغة المستوى العالي إلى لغة الآلة) لتسريع عملية البرمجة ولصعوبة التعامل مع اللغات المتدنية المستوى وذلك باستخدام تعليمات تقوم بالعديد من المهام الجوهرية ، وأهم هذه اللغات : سي و سي بلص بلص C++, C , بيسك Basic , باسكال pascal, فورتران Fortran, كوبول Gobol , ولغة جافا Java .

4. لغات البرمجة المرئية :

هي أحدث اللغات البرمجية التي تعتمد على اللغات الشيئية الموجهة (Object Oriented Languages) والكائنات الجاهزة وأهم هذه اللغات : فيجوال سي بلص بلص Visual c ++ , فيجوال بيسك Visual Basic, فيجوال فوكس برو Visual Fox Pro, دلفي Delphi .

* مقدمة في البرمجة بلغة ++ C:

تستخدم لغة ++Cالأسلوب المهيكل والمنهجي لعملية تصميم البرامج ، حيث تتألف برامج هذه اللغة من مكونات تسمى الصفوف classes والتوابع Functions وبالتالي يمكن تقسيم عملية تعلم لغة الـ++C إلى قسمين : يعتمد الأول منها على تعلم لغة الـ++C نفسها في حين يسمح الثاني بتعليم كيفية استخدام الصفوف الملحقة بهذه اللغة واستخدام التوابع الموجودة ضمن المكتبة المعيارية ANSI C.

1. كيفية كتابة برنامج بلغة ++c:

لكتابة برنامج يجب أن نستهل البرنامج بالمكتبات التي تدعم التعليمات الواجب استخدامها و iostream المكتبة التي سنستخدمها الآن هي مكتبة بحيث تمكننا هذه المكتبة من استخدام الأوامر التالية :

cin>> إدخال وتقرأ c in

cout>> إخراج و تقرأ c out

endl نهاية سطر وتقرأ end line

تتعامل هذه ا لغة ++C مع عدد كبير من المكتبات وكل مكتبة لها اكواد خاصة بها و يمكن التعامل مع عدة مكتبات في آن واحد وسنتحدث عن كل مكتبة عندما يتطلب البرنامج ذلك.

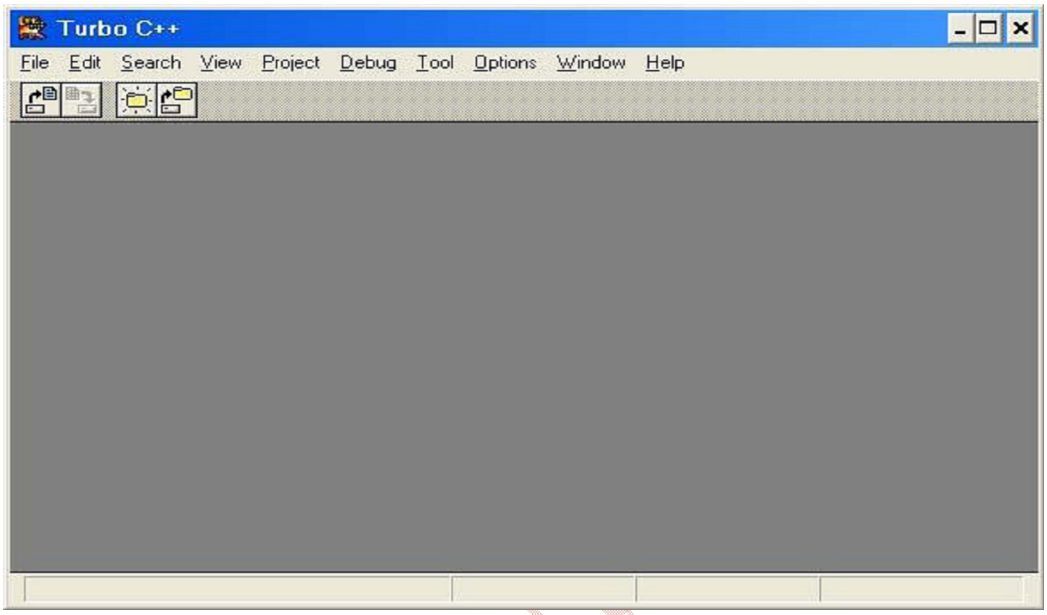
يكتب أي برنامج كما في الشكل التالي:

```
#include <iostream.h>          المكتبات
main( )                        التابع الرئيسي الذي يبدأ من عنده التنفيذ
{                              القوس للدلالة على بداية البرنامج
نبدأ بالإعلام عن المتغيرات
يكتب البرنامج هنا
return 0 ;
}                              القوس للدلالة على نهاية البرنامج
```

وسنستخدم بيئة c turbo++ لبساطتها وسهولة تطبيق البرامج عليها ولأن هدفنا هو تطبيق الخوارزميات التي تعلمناها والتي سوف نتعلمها.

وللدخول إلى بيئة التربو للبحث عن الملف التنفيذي turbo c++ حيث نجده في المجلد : TCWIN45

وسيظهر لنا الشكل التالي :



ولكتابة أي برنامج يكفي أن ندخل إلى القائمة File ونختار منها الأمر New ونبدأ بكتابة البرنامج حيث نقوم بتنفيذه بعد كتابته بالأمر : Run. من القائمة Debug أو بالضغط على(ctrl+F9), ولحفظ البرنامج بعد كتابته ندخل إلى القائمة File ونختار منها الأمر Save as .

2. قواعد تسمية الأسماء التعريفية للبرامج عند حفظها:

1) أن يكون الأسم مكتوباً من سلسلة متصلة من الحروف أو الأرقام بشرط أن يبدأ بحرف أو بخط تحتي " _ ".
2) أن لا يحتوي الأسم رموز خاصة عدا الخط التحتي " _ ".
3) أن لايكون الأسم إحدى الكلمات المحجوزة .

بعض الأمثلة الصحيحة على الأسماء التعريفية للبرامج :

1) B6
2) X_ray
3) Matrix
4) Ok_
5) A
6) Soft_fine
7) Door12
8) _new

بعض الأمثلة غير الصحيحة على الأسماء التعريفية للبرامج للأسباب المبينة إزاء كل منها:

7_up : لانه بدأ برقم وليس بحرف.
b6.1 : لاستعماله الرمز الخاص (.)
salim! : لاستعماله الرمز الخاص (!)
Ъ2 : لا يجوز استعمال حروف غير إنجليزية.
No#1 : لاستعماله الرمز (#)الخاص

ومن الجدير بالذكر أن لغة c++ تفرق بين الحروف الأبجدية الصغيرة والكبيرة ؛ فمثلاً الأسماء: SYSTEM , system, System تعامل كأسماء مختلفة عن بعضها البعض لأن لغة c++ تميز بين الحروف الصغيرة والكبيرة.

3. الكلمات المحجوزة :

وهي كلمات قياسية معروفة مسبقاً لمترجم c++ ، وتكتب عادةً بحروف صغيرة ، ولها معان خاصة بها تؤديها في برنامج c++ ، وهذه بعض الكلمات المحجوزة الهامة :

near	static	asm	double	Long	sizeof
do	int	while	new	Auto	else
for	this	void	delete	Goto	if
const	entry	char	class	Public	case
continue	extern	struct	inline	Float	private
virtual	volatile	frinde	enum	Near	static
cdecl	default	inline	overload	Unsigned	typedef
signed	pascal	operator	switch	Template	union
register	protected	far	catch	Char	const
break	return				

* أمثلة ببسيطة: لتعلم مبادئ أساسية فى لغة ++ C :

1- طباعة نص مؤلف من سطر :

```
// First Program          كل الكتابات التي تلي هذه الإشارة (// ) تسمى تعليق لا يتم تنفيذه
# include<iostream.h> //  الملف الرأسي الحاوي على العمليات الخاصة بالدخل والخرج
main ()                  // التابع الرئيسي الذي يبدأ من عنده التنفيذ
{                        // بداية البرنامج
   cout <<  " welcome to c++ " ;   // تعليمة الطباعة
return 0 ;               // إحدى طرق الخروج من التابع
}                        // نهاية البرنامج
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
welcome to c++
```

*** سلاسل الهروب :** يدعى \ بحرف الهروب وهو يلحق بحرف يدل على معنى معين كما هو موضح في الجدول:

سلسلة الهروب	المعنى
\ n	سطر جديد أي وضع المؤشر في بداية السطر التالي
\ t	تحريك المؤشر مسافة جدولية أفقية (tab)
\"	تستخدم لطباعة علامة الاقتباس

2- طباعة نص مؤلف من سطر مع استخدام حرف الهروب :

```
# include <iostream.h>
main ( )
{
      Cout <<"welcome to c++\n " ;
                                  حرف الهروب
return 0;
}
```

3- أكتب برنامجاً يقوم بطباعة مستطيل من النجوم:

```
# include < iostream.h>
main ( )
{
        cout << " ***********\n" ;
        cout << " *\t " <<" *\n";
        cout << " *\t " <<" *\n";
        cout << " *\t " <<" *\n";
        cout << "***********\n";
return 0;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
***********
*         *
*         *
*         *
***********
```

4 - برنامج جمع عددين صحيحين :

```
# include < iostream.h>
main ( )
{
   int  x1 , x2, x3 ;                          // تعريف المتحولات
     cout <<" enter first numbe ";          // تعليمة الطباعة
     cin >> x1 ;                                  // تعليمة قراءة متحول
     cout << " enter second number ";
     cin >> x2 ;
     x3 = x1 + x2 ;                  //x3 إجراء عملية الجمع والإسناد إلى المتحول الجديد
    cout << "sum is " <<x3 ;              // تعليمة الطباعة المتعددة
return 0 ;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

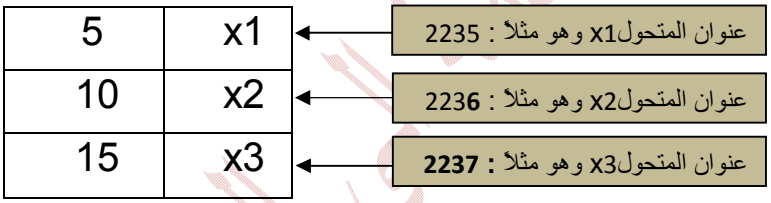
```
enter first number  10
enter second number  55
sum is  65
```

4. المتحولات (المتغيرات) :

على تسمية المتحولات نطبق نفس قواعد تسمية الأسماء التعريفية للبرامج مع أفضلية تسمية المتحولات بأسماء تعبر عن مضمونها.

ملاحظة: مفهوم يتعلق بالذاكرة ألا وهو طريقة حجز المتحولات:

كل اسم من أسماء المتحولات مثل .... , x3 , x2 ,x1 يتم وضعه في الذاكرة ويعرف بإسم name ونمط type وحجم size وقيمة value وبالتالي فإن المتحول x1 يملك الاسم x1 والنمط int والحجم 4 بايت والقيمة هي حسب القيمة المقروءة .



مواضع المتحولات في الذاكرة مع ذكر الاسم والقيمة

1. أنواع المتحولات:

- المتحول المنطقي Boolean
- المتحول المحرفي char
- المتحولات الصحيحة short int, int , long int , unsigned int
- المتحولات الحقيقة float , double , long double

ويبين الجدول التالي أنواع المتحولات ومجالاتها :

نوع المتحول	الحجم	المجال
char	1 byte	-128 to 127
int	4 bytes	-2147483648 to 2147483647
short int	2 bytes	-32768 to 32767
Long int	4 bytes	-2147483648 to 2147483647
bool	1 byte	True or false
float	4 bytes	-3.4E-38 to 3.45E+38
double	8 bytes	-1.7E-308 to 1.7E+308
long double	8 bytes	-1.7E-308 to 1.7E+308

تقوم C++ بتطبيق العمليات في العبارات الحسابية حسب ترتيب معين محدد تبعاً لقواعد الأولوية بين العمليات التي تماثل قواعد الأولوية في الجبر وذلك كما في الجدول التالي:

2. **العمليات الحسابية** :

اسم العملية	الرمز الحسابي	طريقة التعبير حسب لغة C++
الجمع	+	x1 + x2
الطرح	-	x2 – x1
الضرب	*	x1 * x2
القسمة	/	x1 / x2
باقي القسمة الصحيحة	%	x1 % x2

أما بالنسبة لعمليتي الإسناد والمقارنة فتتم بالشكل التالي:

الشكل الجبري	الشكل الموافق حسب c++	مثال	معنى الكتابة
=	= =	x = = y	x تساوي y
≠	! =	x != y	x لا تساوي y
<	<	x <y	x أصغر من y
>	>	x >y	x أكبر من y
≤	<=	x < = y	x أصغر أو يساوي y
≥	>=	x > = y	x أكبر أو يساوي y

3. **العملية المنطقية Logical operators :**

وهي ثلاثة :

and يرمز لها &&

or يرمز لها ||

not يرمز لها !

*** بعض الأمثلة:**

1. أكتب برنامجاً يأخذ كدخل ثلاث أعداد صحيحة من لوحة المفاتيح ثم يطبع مجموعها ومتوسطها الحسابي.

```
# include < iostream.h>
main ( )
{
int a , b, c ;
cout << " enter a =" ; cin >> a ;
cout << " enter b = " ; cin >> b ;
cout << " enter c = " ; cin >> c ;
cout << " sum is :  " << a+b+c << " \ n" ;
cout << average is :  " << ( a+b+c)/3 <<" \n";
return 0 ;
}
```

```
enter   a = 10
enter   b = 20
enter   c = 33
sun is  63
average is      21
```

2. أكتب برنامج يقرأ (يدخل) نصف قطر دائرة ثم يطبع قيمة محيطها و مساحتها.

ملاحظة : قيمة π =3.14

```
# include <iostream.h>
main  ( )
{
float   r,s,m ;                          // تعريف متحول حقيقي
float  p = 3 , 14 ;              // تعريف متحول حقيقي وإسناد قيمة له
cout << " enter  r =" ; cin >> r ;
s=2*p*r;
m= p*r*r;
cout <<" s=  " << s<<"\n" ;
cout << " m = " << m ;
return 0 ;
}
```

```
enter  r = 4.5
s = 28.26
m = 63.585
```

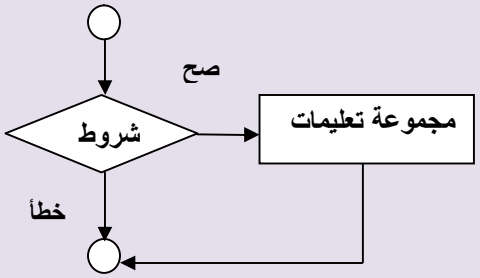
iii. بنى التحكم (Control Structures)

1. البنى الشرطية:

a) بنية الأختيار الشرطية if :

تقوم بنية الاختيار if بتنفيذ فعل معين عندما يكون الشرط المرافق لها محققاً وإلا يتم تجاهلـــه ، ولها الشكل العام التالي :

```
if (شروط )      تعليمة1 ;
أو
if (شروط )
{
تعليمة1 ;
تعليمة2 ;
}
```



مثال 1: علامة النجاح في أحد الامتحانات تساوي 60 درجة عندها فإن تعليمة الـــ if تكون بالشكل:

```
if  ( grad > = 60 )  cout <<"passed";
```

مثال 2: أكتب برنامجاً يطلب من المستخدم إدخال عددين صحيحين. ثم يقارن بينهما ؟

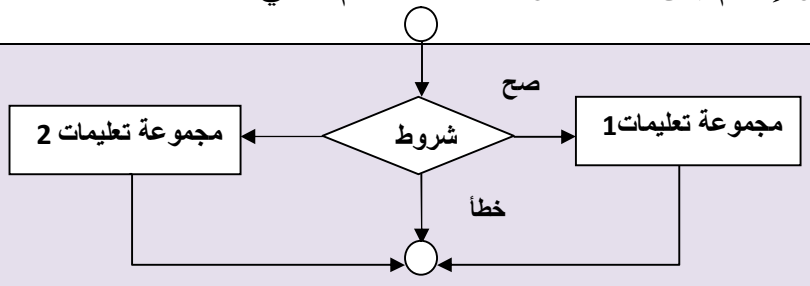
```
#include <iostream.h>
main ()
{
        Int a, b;
        cout<<" a=";cin>>a;
        cout<<" b=";cin>>b;
        if ( a > b ) cout <<a<< " >"<< b ;
        if ( a < b ) cout <<a<<" < "<< b ;
        if ( a = = b ) cout <<a<<"="<< b;
        return 0 ;
}
```

```
a = 100
b = 69
100 > 69
```

بنية الأختيار الشرطية if/else

تسمح بنية الاختيار if / else بتحديد جملة من الأفعال الممكن تنفيذها إذا كان الشرط المرافــق صحيحاً أو إذا لم يكن كذلك ، ولها الشكل العام التالي:

```
if ( شروط )
    تعليمات1 ;
else
    تعليمات2 ;
```



مثال 1: إذا كان علامة الطالب أكبر أو يساوي القيمة 60 درجة فيطبع كلمة "passed" وإلا فهي تطبع الكلمة "failed" .عندها فإن تعليمة الـــ if / else تكون بالشكل:

```
if ( grad > = 60 )
        cout << " passed " ;
else
        cout << "failed" ;
```

مثال 2: أكتب برنامجاً يقرأ عدداً صحيحاً ثم يحدد و يطبع فيما إذا كان هذا العـــدد زوجيـــاً أم فردياً .

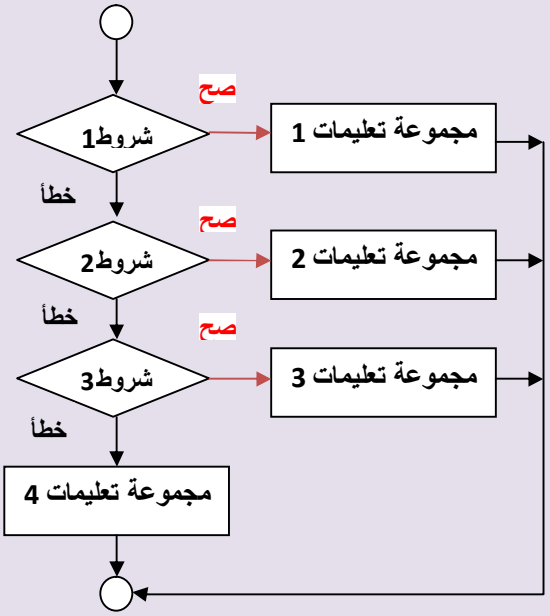
```
# include < iostream.h>
main ()
{
        int a ;
        cout <<"enter a ="; cin>>a;
        if ( a % 2 = = 0)
                cout << " not odd" ;
        else
                cout << " odd" ;
        return 0 ;
}
```

```
enter a = 13
odd
```

b) البنى الشرطية المتعددة :

ويمكن استخدام البنى الشرطية if / else المتعددة من أجل القيام بفحص عدة حالات من خـــلال وضع البنى if / else تحت بعضها البعض , ولها الشكل العام التالي:

```
if ( شروط 1 )
        تعليمات1 ;
else if  ( شروط 2 )
        تعليمات2 ;
else if  ( شروط 3 )
        تعليمات3 ;
else
        تعليمات4 ;
```



مثال: ادخل علامة الإمتحان من 100 فيتم طباعة الحرف المقابل لهذه العلامة:

```
if ( (grad >= 90)&&(grad)<=100)
        cout << "A" ;
else if ( (grad >= 80)&&(grad)<90)
        cout << "B" ;
else if ( (grad >= 70)&&(grad)<80)
        cout << "C" ;
else if ( (grad >= 60)&&(grad)<70)
        cout << "D" ;
else if ( (grad >= 0)&&(grad)<60)
        cout << "H" ;
else
        cout << "error" ;
```

ملاحظة : عادة تضع تعليمة واحدة في جسم البنية الاختيارية if ولكن إذا أردنا وضع عدة تعليمات يجب أن نقوم بوضعها داخل قوسين كبيرين ({ }) . نسمى مجموعة التعليمات المحتواه ضمن زوج من الأقواس الكبيرة بالتعليمية المركبة compound statement .

مثال:

```
if (grad > = 60 )
        cout << " passed" ;
else
        {
        cout << " failed " ;
                cout << " you must take this course again" ;
        }
```

في هذه الحالة إذا كانت قيمة grad أصغر من 60 عندها يقوم البرنامج بتنفيذ التعليمتين الموجودتين في الجزء else ويطبع ما يلي:

failed

you must take this course again

بعض الأمثلة:

1) أكتب برنامج يأخذ كدخل عددين صحيحين من لوحة المفاتيح ويفحص فيما إذا كـــان الثاني قاسم للأول.

```
# include<iostream.h>
main ( )
{
        int  a , b ;
        cout<<"enter a=";cin>>a;
        cout<<"enter b=;cin>>b;
        if ( (b! = 0) && (a % b = = 0) )
                cout << a << " is divisible by " <<b ;
        else
                cout <<a<<" is not divisible by " << b ;
        return 0 ;
}
```

```
enter a = 25
enter b = 5
25 is divisible by 5
```

2) أكتب برنامج يأخذ كدخل ثلاث أعداد صحيحة ثم يطبع أصغر هذه الأعداد.

```
# include < iostream.h>
main ()
{
int a , b, c ;
cout << " a= ";  cin >> a;
cout << "b= ";  cin >> b;
cout << " c= ";  cin >> c;
if (( b > a)&&(c > a) )
                    cout << " min is" << a ;
else   if (( a > b)&&(c > b))
                     cout << " min is " << b ;
else
                     cout << " min is " << c ;
return 0;
}
```

```
a = 10
b = 8
c = 77
min is  8
```

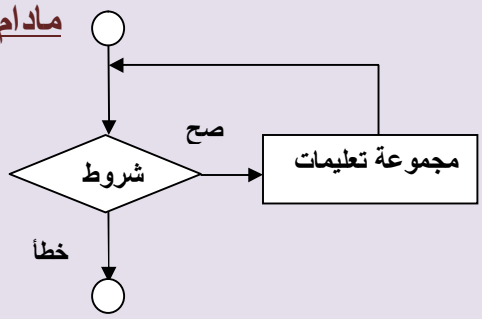
2. البنى التكرارية :

a) البنية التكرارية While :

تسمح البنية التكرارية للمبرمج بتحديد مجموعة من الأفعال يجري تكرارها طالما ظـــل الشـــرط المرافق للبنية محققاً ، ولها الشكل العال التالي :

```
while (شروط )
{
تعليمة1 ;
تعليمة2 ;
}
```

مادام (الشروط محققة) كرر



مثال 1: أكتب برنامج لطباعة الأعداد من 1-10 بشكل عمود واحد.

```
# include < iostream.h>
main ()
{
      int  i=1 ;
      while ( i <=10)
      {
            cout << i << "\n" ;
            i = i +1 ;
      }
      return 0 ;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
1
2
3
4
5
6
7
8
9
10
```

مثال 2: يفرض لدينا علامات مذاكرة قام بها طلاب صف مؤلف من عشرة طلاب والمطلــوب حساب معدل علامات طلاب الصف في هذه المذاكرة .

```
# include < iostream.h>
main ()
{
      float   mark , sum ;
      int  i = 1;
      sum = 0;
      while ( i < = 10 )
      {
            cout<<"enter the mark=";    cin>>mark;
            sum = sum + mark ;
            i = i +1 ;
      }
      cout<<"average is : "<<sum/10 ;
      return 0 ;
}
```

نتائج البرنامج التى تظهر على الشاشة السوداء هى :

```
enter the mark = 13
enter the mark = 44
enter the mark = 54
enter the mark = 60
enter the mark = 90
enter the mark = 33
enter the mark = 75
enter the mark = 56
enter the mark = 55
enter the mark = 78
average is : 55.8
```

ملاحظة:

1. إن عدم وضع تعليمة أو فعل جسم البنية while يسبب عدم تحقق الشرط المرافق لها وينتج عن ذلك عدم إنتهاء التكرار .

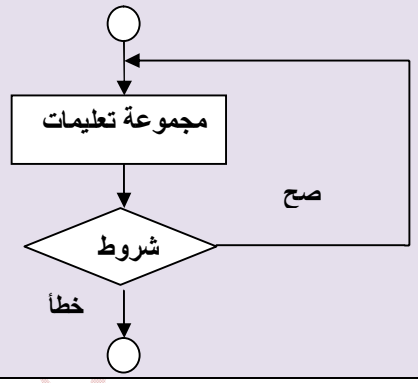
2. تسبب كتابة الكلمة while مع حرف كبير في البداية خطأ وذلك على اعتبار أن لغة C++ حساسة لحالة الحروف تحتوي كافة الكلمات المفتاحية الخاصة بلغة C++ مثل if , while ، .. وغيرها على شكل حروف صغيرة .

3. إن أي متحول لا يعطي قيمة ابتدائية يمكن أن يكون له قيمة ما لا يعرف عنها شيء مخزنة مسبقاً في موضع الذاكرة المخصص لهذا المتحول ، وبالتالي إن عدم إعطاء متحول حساب مجموع مثل sum أو عداد مثل i سوف يؤدي إلى الحصول على نتائج قد تكون خاطئة.

b) البنية التكرارية do / while :

تشبه بنية التكرار do / while البنية while حيث نقوم بالبنية while بالتحقق من صحة شرط الاستمرار بالتكرار في بداية الحلقة قبل تنفيذها ، أما في حالة البنية do / while فيتم ذلك بعد تنفيذ جسم الحلقة أولاً . أي يتم تنفيذ جسم البنية do / while مرة واحدة على الأقل. عند الإنتهاء من تنفيذ البنية do / while يتم الانتقال إلى التعليمية التي تليها مباشرة ، ولها الشكل العام التالي:

```
do {
    تعليمة1 ;
    تعليمة2;
  }
while (شروط) ;
```



وقد تم استخدام الأقواس الكبيرة لتحديد جسم البنية do / while حتى لا يتم الخلط بين البنيتين while , do / while .

مثال 1: نفس المثال السابق ــ أكتب برنامج لطباعة الأعداد من 1–10 بشكل عمود واحد ولكن باستخدام do / while :

```
# include <iostream.h>
main()
{
      int  i =1;
      do {
            cout <<i<<"\n";
            i=i+1;
         } while ( i <=10);
      return 0;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
```

c)بنية التكرار for:

تدعى هذه البنية أيضاً بالبني التكرارية ذات العداد ، وهي تتطلب ما يلي:

1- تعريف متحول التحكم بالحلقة (وهو عداد الحلقة)

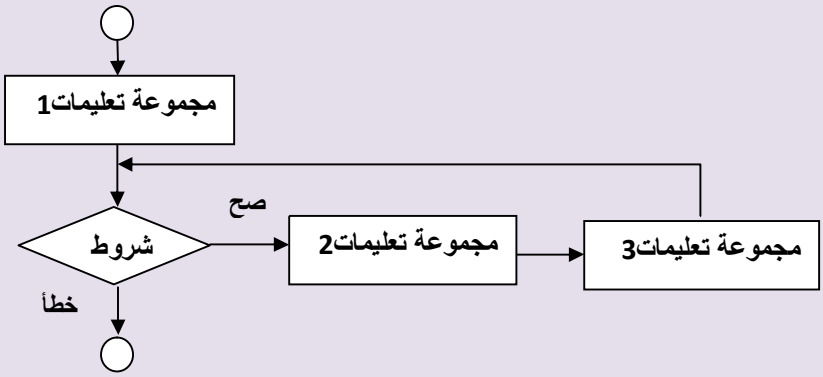
2- تحديد القيمة الابتدائية لمتحول التحكم بالحلقة .

3- تحديد أسلوب الزيادة (أو الانقاص) الذي يتم من خلاله تغيير قيمة متحول التحكم بالحلقة في كل مرة نمر فيها.

4- تحديد الشرط الذي من خلاله نقوم بفحص النتيجة النهائية لمتحول التحكم بالحلقة (حتـــى نحدد إذا كان من الممكن معاودة تنفيذ الحلقة).

ولها الشكل العام التالي:

```
for ( exp 1; exp 2; exp 3 )
{
تعليمة1 ;
تعليمة2 ;
}
```



exp 1 : يمثل تعريف وتحديد القيمة الابتدائية لعداد الحلقة.

exp 2 : يمثل شرط إنهاء الحلقة أي شرط فحص النتيجة النهائية لعداد الحلقة.

exp 3 : يمثل أسلوب زيادة أو إنقاص عداد الحلقة.

مثال1: نفس المثال السابق ــ أكتب برنامج لطباعة الأعداد من 1- 10 بشكل عمــود واحــد ولكن بإستخدام البنية for .

```
# include <iostream.h>
main()
{
      for ( int i= 1; i<=10 ; i=i+1)
            cout <<i<<"\n";
      return 0;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
1
2
3
4
5
6
7
8
9
10
```

مثال 2: أكتب برنامج لحساب مجموع جميع الأعداد الصحيحة من 2 إلى 100

```
# include < iostream.h>
man ( )
{
        inst sum = 0 ;
        for ( int i = 2 ; i < = 100 ; i = i +1)
                sum  = sum + i ;
        cout << " sum is  " << sum ;
        return 0 ;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
        Sum is   5049
```

مثال3 : أكتب برنامج يلخص نتائج امتحان مادة ما لعشرة طلاب وذلك بعد أن أعطيت قائمة بأسماء الطلاب ومقابل كل اسم تم وضع القيمة **1** إذا كان الطالب ناجح والقيمة **0** إذا كان الطالب راسب في الامتحان .

```
# include < iostream.h>
main ( )
{
      int  r , p, f ;
      p = 0 ; f = 0 ;
      for ( int i = 1 ; i < = 10 ; i ++ )
      {
            cout << " enter result : "; cin >> r ;
            if ( r = = 1 )
                  p  = p+1 ;
            else
                  f=f+ 1 ;
      }
      cout << " passed : " <<p <<"\n" ;
      cout << " failed : ' << f << "\n" ;
      return 0 ;
}
```

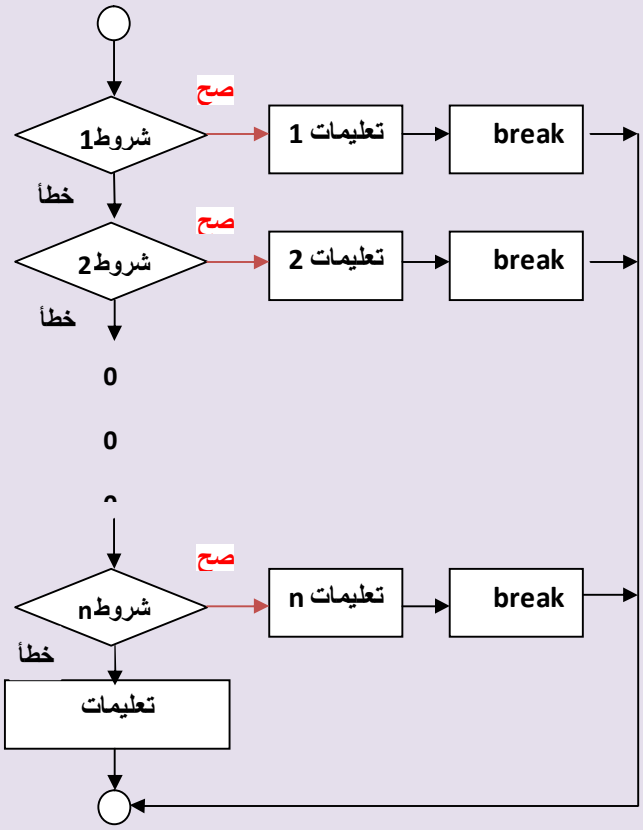
نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
enter result   : 1
enter result   : 1
enter result   : 1
enter result   : 0
enter result   : 1
enter result   : 0
enter result   : 0
enter result   : 1
enter result   : 1
enter result   : 0
passed : 6
failed  : 4
```

3.بنية الاختيار المتعدد switch :

يمكن أن تصادفنا حالة خاصة في إحدى البرنامج تحتوي على سلسلة من القرارات التي تتعلق بنتائج متعدد لفحص قيمة متحول أو تعبير ما ، ويمكن أن تؤدي كل نتيجة من هذه النتائج إلى القيام بفعل مختلف عن الآخر . لذلك توفر لغة C++ البنية switch من أجل التعامل مع حالات اتخاذ القرار المتعلقة بعد اختيارات ، ولها الشكل العام التالي:

```
switch (شروط)
{
case  1ثابت : 1تعليمة ;
case  2ثابت :  2تعليمة ;
case  3ثابت : 3تعليمة ;
case  4ثابت : 4تعليمة ;
.
.
.
case  n ثابت : n تعليمة ;
default : 0 تعليمة ;
}
```



مثال 1:أكتب برنامج لإعطاء اسم اليوم من أيام الأسبوع عند إعطاء رقمه.

```
# include < iostream.h>
main ()
{
	int c ;
	cout << "enter number : " ;
	cin >> c ;
	switch (c )
{
	case 1 : { cout << " saturday " ; break ; }
	case 2 : { cout << " sunday " ; break ; }
	case 3 : { cout << " monday " ; break ; }
	case 4 : { cout << " tuesday " ; break ; }
	case 5 : { cout << " wednesday " ; break ; }
	case 6 : { cout << " thursday " ; break ; }
	case 7 : { cout << " friday " ; break ; }
	default : { cout << " that number is out of range " ; }
}
	return 0 ;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
	enter number    : 7
	friday
```

مثال2: أكتب برنامج يقوم بقراءة عددين ومن ثم يعطي ناج جمعهـا وطرحهمـا وضـربهما مستخدماً لعرض ذلك شاشة خيارات.

```
# include < iostream.h>
main ( )
{
      int  n , x, y ;
      cout << "1:  جمع العددين " ;      cout <<"\n";
      cout << "2 :  طرح العددين " ;    cout <<"\n";
      cout <<3:  ضرب العددين" ;      cout <<"\n";
      cout << "*******************" ;      cout <<"\n";
      cout << " أدخل العدد الأول " ; cin >>x; cout <<"\n";
      cout << " أدخل العدد الثاني " ; cin >> y ; cout <<"\n";
      cout <<  "أدخل رقم الخيار" ; cin >> n ; cout << "\n";
while ( n!=0)
     {
          switch ( n )
          {
           case 1:
                { cout << x+y ; break ; }
            case 2:
                { cout << x-y ; break ; }
           case 3 :
                ( cout << x*y; break; }
           default :
               { cout << "  الرجاء إدخال أحد أرقام الخيارات المتاحة" ; cin>>n;}
           }
     }
return 0;
}
```

أمثلة عامة:

1-أكتب برنامج لقراءة ثلاث أعداد a, b, c ثم التحقق هل تصلح هذه الأضلاع لأن تكـــون أضلاع مثلث أم لا ، وبمعنى آخر هل يمكن أن نجد مثلث أطوال أضلاعه هي a, b, c .

```
# include < iostream.h>
# include < math.h>
    // int abs (int) الملف الرأسي الحاوي على جميع التوابع الرياضية وتم استخدامه من أجل التابع
main ( )
{
      int  a , b, c ;
      cout << " a : " ; cin >> a ;
      cout << " b : " ; cin >> b ;
      cout << " c : " ; cin >> c ;
      if ((a+b>c) && (abs(a-b)<c)&&(b+c>a)&&(abs(b-c)<a) &&(a+c>b) &&
(abs (a-c)<b))
      cout << " triangle " ;
else
      cout << " not triangle " ;
return 0;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
a : 5
b : 4
c : 3
triangle
```

2- أكتب برنامج لحساب n!

```
# include < iostream.h>
main ( )
{
      Int  n ;
      double  fact = 1 ;
      cout << " enter  value  n: " ; cin >> n;
      if ( n = = 0 )
            cout << " n! = 1;
      else
      {
            for ( int i = 1 ; i < = n ; i ++)     //  i++   هي نفس  i=i+1
            fact=fact * i  ;
            cout << " n ! = " << fact ;
      }
      return 0 ;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
enter  value  n : 5
n ! = 120
```

3- برنامج إيجاد قواسم عدد X

الحل: إذا فرضنا أن العدد 30= x فإنا نختبر الأعداد التي قبل x بحيث إذا كان باقي القسمة عليها يساوي الصفر عندئذ يكون العدد قاسما للعدد x .

```
# include < iostream . h >
main ()
{
      int  x  ;
      cout << " enter number : " ; cin >> x ;
      for ( int i = 1 ; i < =x ; i ++)
      if ( x % i = = 0 )
            cout << i << " \n";
      return 0 ;
}
```

نتائج البرنامج التى تظهر على الشاشة السوداء هى :

```
enter number : 30
1
2
3
5
6
10
15
30
```

4. برنامج يقوم بقراءة عدد ما x ومن ثم يحدد هل هذا العدد أولي أم لا.

ملاحظة للحل :

1- لا يوجد في لغة الـــC++ نمط بولياني لذلك ننشئ نمط من خلال النمط التعدادي enum.

2- لحل هذه المسألة يلزمنا متحول اختبار f من نوع Boolean ففي البداية نسـند القيمـة false إلى هذا المتحول أي نفرض أن العدد ليس أولي ، ومن ثم نبحث هل هناك عدد يقسـم x وفي حال وجوده نسند لـــf القيمة True . وفي النهاية نختبر قيمة المتحول f وأعتمـاداً عليـه نحدد هل العدد أولي أم لا.

```
# include < iosteam.h >
enum boolean {true, false };                    // التصريح عن نمط تعدادي
main ( )                    {
boolean  f = false ;
int x ;
cout <<' enter number: " ; cin >> x ;
for ( int i = 2 ; i <x ; i ++)
        if ( x % i = = 0 )
        f = true ;
if ( f = = false )
        cout << " the x number is primary " ;
else
        cout << " the x numbe is not primary " ;
return 0 ;
    }
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
enter number : 67
the  x number is primary
```

4- أكتب برنامج لحساب الحدود العشرة الأولى لهذه السلسة :

$$z = 1 - \frac{1}{1} + \frac{1}{2} - \frac{1}{3} + \frac{1}{4} - \ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

```
# include < iostream.h>
# include < math.h>
main ( )
{
        int  n ;
        float z = 1;
        cout << " enter n: ' ; cin >> n;
        for ( int i = 1 ; i <n ; i ++ )
        if ( i % 2 = = 0 )
                z+= pow(i , -1) ;          // تابع الرفع لقوة ويوجد في الملف math
        else
                z-=pow (i , -1 ) ;
        cout << " z = " << z ;
        return 0 ;
}
```

نتائج البرنامج التى تظهر على الشاشة السوداء هى :

```
        enter  n : 15
z = 0.341295
```

5- أكتب برنامج لإيجاد القاسم المشترك الأعظم لعددين وذلك باستخدام طريقة إقليدس التي تتلخص كما يلي: أقوم بطرح العدد الأصغر من العدد الأكبر وأجعل حاصل الطرح مكان الأكبـــر حتى تصبح القيمتين متساويتين فتكون قيمة التساوي هذه هي القاسم المشترك الأعظم GCD .

مثال : العددين 15 و 20

20	15
5	15
5	10
5	5

القاسم المشترك الأعظم


```
#include < iostream.h>
main ( )
{
      int x , y ;
      cout << "enter x : " ; cin >> x ;
      cout << " enter y : " ; cin >> y;
      while ( x! = y )
      {
if ( x > y )
      x - = y ;
else
      y - = x ;
}
cout << " the gcd is " << x ;
return 0 ;
 }
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
enter x : 10
enter y : 35
the gcd is 5
```

6- أكتب برنامج لقراءة n عدد ثم حساب مجموع هذه الأعداد ومتوسطها وأكبر وأصغر عدد فيها:

ملاحظة: دائماً لحساب أكبر أو أصغر عدد من بين مجموعة أعداد ، نفرض أن العدد الأول هو الكبير ثم نختبر باقي الأعداد وكلما ظهر عدد أكبر جديد نجعله هو العدد الأكبر ، وهكذا حتى تنتهي مجموعة الأعداد . (بالنسبة للعدد الأكبر).

```
# include < iostream.h>
main ( ) {
        int  n , x , sum , max , min ;
        cout << " enter n : " ; cin >> n;
        cout << " enter the first number : " ; cin >> x ;
        sum = x ; min = x ; max = x ;
        for ( int i = 2 ; i < = n ; i ++)
        {
                cout << " enter number : " ; cin >> x ;
                sum  = sum+x ;
                if ( x > max )     max = x ;
                if ( x < min )      min = x ;
        }
        cout << " sum is " << sum << "\n" ;
        cout << " avg is " << ( float ) sum /n << "\n" ;
        cout << " max is " << max << " \n" ;
        cout << " min is " << min << "\n" ;
        return 0 ;  }
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
enter  n : 4
enter the first number : 22
enter number : 13
enter number : 24
enter number : 44
sum is 103
avg is 25.75
max is 44
min is 13
```

7- أكتب برنامج لقراءة عدد ما والتحقق فيما إذا كان عدم تام أم لا .

الحل :

نقول عن عدد ما أنه عدد تام إذا كان مجموع قواسم هذا العدد (ما عدا العدد نفسه) يساوي العدد نفسه .

مثال : العدد 6 هو عدد تام لأن مجموع قواسم العدد 6 تساوي 06 (6=3+2+1)

```
# include <iostream.h>
main ( )
{
      int x ;
      int sum = 0 ;
      cin>> x ;
      for ( int  i = 1 ; i < x ; i ++)
      if ( x % i = = 0 )
            sum + = i ;
      if ( sum = = x )
            cout << " perfect " ;
      else
            cout << " not perfect " ;
      return 0 ;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
enter number = 28
perfect
```

8- أكتب برنامج لإيجاد جميع الأعداد التامة ضمن مجال [1..n]

```
# include < iostream.h>
main ( )
{
      int n , sum = 0 ;
      cin>>n ;
      for ( int i = 1 ; i <=n ; i ++ )
      {
            for ( int j = 1 ; j < i ; j++)
                  if ( i % j = = 0 )
                        sum + = j ;
                  if ( sum = = i )
                        cout << "  " <<i << endl;
                  sum = 0 ;
            }
      return 0;
      }
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
enter  n : 200
 6
 28
```

9- أكتب برنامج لإيجاد المضاعف المشترك الأصغر لعددين:

```
# include < iostream.h>
main ( )
{
        int x , y ;
        cout << " x = " ; cin >> x;
        cout << " y = " ; cin >> y ;
        if ( x > = y)
        {
                for ( int j = x ; j < x ; j++)
                        if ( j % x = = 0 ) && ( j % y = = 0 )
                                { cout <<j ; break ; }
    }
else
    {
        for  ( int j = y ; j < x *y; j++)
                if (( j% x = = 0) && ( j % y = = 0 ))
                        { cout << " " << j ; break ; }
   }
return 0 ;
  }
```

10- أكتب برنامج لقراءة عددين والتحقق فيما إذا كانا عددين صديقين أم لا .

الحل : نقول عن عددين أنهما صديقين إذا كان مجموع قواسم العدد الأول (ما عدا العدد نفســـه) يساوي العدد الثاني والعكس بالعكس.

```
# include<iostream.h>
main ( )
{
   int x , y , i ;
    int sum 1 = 0 , sum 2=0
       cout <<"x="; cin>>x;
       cout <<" y=" ; cin >> y;
       for ( i = 1 ; i < x ; i ++ )
           if ( x % i = = 0 )
              sum 1 + = i ;
       for ( i = 1 ; i < y ; i ++)
            if ( y % i  = = 0 )
              sum 2 + = i ;
       if ( sum 1 = = y && sum 2 = = x )
              cout <<x<<" friend " <<y;
       else
              cout << x << " not friend " << y;
       reterun 0 ;
}
```

نتائج البرنامج التي تظهر على الشاشة السوداء هي :

```
x = 20
y = 34
20 not  friend  34
```

وظيفة :

أكتب برنامج لإيجاد جميع الأعداد الصديقة ضمن مجال [1.. n] .

-

مع تمنياتي با لتوفيق

مدرس المقرر

د. عميد صالح غازي

يمكن التواصل و إبداء أرائكم وإقتراحاتكم عن طرق الموقع الشخصي والبريد الإلكتروني التاليين:

www.faculty.ksu.edu.sa/dr-ameed

aghazi@ksu.edu.sa