أسئلة المراجعة

١. ما هو أقل عدد من التكرار

أ. يمكن أن يتم بالحلقة التكرارية while؟

ب.يمكن أن يتم بالحلقة التكرارية do…….while؟

٢. ما الخطأ في الحلقة التكرارية الآتية :

```
while (n < = 100)
sum += n*n;
```

٣. اذا كانت sجملة مركبة و e1و e2و e3 تعبيرات فما الفرق بين جزئية البرنامج التالية:

```
for (e1; e2 ;e3 )
s;
```

وجزئية البرنامج التالية:

```
el;
while (e2) {
s;
s3;
}
```

٤. ما الخطأ في البرنامج التالي:

```
main()
{
const double pI ;
int n;
pI =3.14159265358979
n=22;
n=22;
}
```

٥. ما المقصود ب تكرار لا نهائي "infinite loop" وكيفية الاستفادة منه؟

٦. كيف يمكن بناء حلقة تكرارية بحيث تنتهي بجملة داخل البلوك الخاص بها؟

٧. لماذا يجب تجنب اختيارات التساوي مع المتغيرات ذات النقطة العشرية المتنقلة؟

مسائل محلولة

٨. تتبع الجمل الآتية مبينا قيمة كل متغير في كل مرة

```
int x, y, z;
x= y =z =6;
x * =y  + = z - = 4;
```

في البداية تم تخصي العدد ٦ لكل من المتغيرات zو y, x بعد ذلك نقص المتغير zبمقـدار 4فأصبحت قيمته 2 بعد ذلك زادت قيمة y بمقدار ٢ فأصبحت قيمته 8عند ذلك ضرب xفي ٨ فأصبحت قيمتها ٤٨ .

٩.٣ نفترض أن e هى تعبير و s هى جملة و المطلوب تحويل كل من الحلقات التكراريـة for الأتية إلى ما يناظرها من الحلقات التكرارية while.

a. for (; e ;) s
b. for (; ; e) s
c. while (e) s:
b. while (1) (s; e ;).

بفرض أن s لا تحتوي على عبارة continue(انظر المسألة ٣.٣)

١٠. حول الحلقة التكرارية for إلى حلقة تكرارية while

```
for ( int i=1 ; i<= n ; i++)
     cout << i*I;
int  i = 1;
while (i <= n)
      cout << i*i;
       i++;
}
```

١١. صف شكل الخرج من البرنامج التالي:

```
main ()
{
for (int  i= 0;  i< 8 ; i++)
     if (i%2 = = 0)cout << i +1<<end1;
     else if (i%3 = = 0) cout i*i<<endl;
     else  if (1%5= =) cout << 2*i – 1<< end;
      else cout << i<<endl;
}
```

1 1 3 9 5 9 7 7

١٢.صف شكل الخرج من البرنامج التالي:

```
main ()
{
```

```
  for (int i=0; i<8; i++){
        if (i%2= = 0) cout << i+1<< endl;
         else if (i%3= = 0) continue;
         else if (i%5= = 0) break ;
         cout << "end of program .\n";
}
         cout << "end of program \n";
}
```

```
1
end  of program
end  of program
3
end  of program
5
end  of program
end  of program
```

١٣. في النوع float الذي يحتوي على ٢٣ بت يتم استخدامها لتخزين الجزء العشـري و ٨ بت تستخدم لتخزين الأس.

أ. كم عدد الخانات الناتجة عن النوع float يحتوي على ٣٢ بت؟

ب. ما مدى المقدار الذي يحتوي على ٣٢ بت من النوعfloat؟

ج أ. الـ٢٣ بت تبدأ من البت رقم ٢ إلى البت رقم ٢٤ في الجزء العشري . البت الأولـي يجب أن تكون ١ لذلك فانها غير مخزنة وهكذا فان ٢٤ بت تم تمثيلهم هذه الـ٢٤ بت يمكن أن تحفظ $٢^{٢٤}$ من الأرقام الـ 2^{24}=16.777.216تمثل بـ ٧ خانات في المدي الكامل لذلك فـان الـ٧ خانات يمكن تمثيلهم . لكن أخر بت يكون هناك شك فيها بسبب التقريـب وعلـى ذلـك فالنوع floatذو ٣٢ بت ينتج عنه ٦ خانات للدقة.

ج ب. الـ٨ بت المستخدمة في الأس يمكن أن تحتفظ ب 256 $=2^8$ عدد مختلف . اثنين مـنهم محجوزين لبيان تجاوز الحد الأقصى والحد الأدنى المسموح بهما يتبقى ٢٥٤ عدد للأس لـذلك فان مدى الأس يكون من ١٢٦ إلى 127+ينتج مقدار في المدى من -10 x 1.175494=2^{-126} إلى38 10^{38} x 1.70141 =$٢^{١٢٧}$

مسائل محلوله في البرمجة

١٤. اكتب برنامج يحول البوصاتinchesإلى سنتيمتراتcentimeters على سبيل المثال لـو أن المستخدم ادخل ١٦.٩ بوصة تعبر عن الطول فان الخرج يكون ٤٢.٩٢٦ سـم (البوصـة الواحدة = ٢.٥٤سم)

سوف تستخدم متغيرين من النوع float.

```
main( )
{
float inches, cm;
cout <<  " Enter Length in inches : ";
cin >> inches;
cm = 2.54*inches;
cout << inches <<  " inches =  " << cm <<  " centimeters .\n ";
}
Enter length in inches: 16.9
16.9 inches = 42.926 centimeters
```

البرنامج ببساطة يقرأ الدخل للمتغير inches ويحوله إلى سـنتيمترات ويطبـع النـاتج فـي المتغيرcmويخرجها.

١٥. اكتب برنامج لإيجاد الجذر التربيعي الصحيح لأي عدد معطي (الجذر التربيعي هو العـدد الصحيح الذي يكون تربيعه أقل من أو يساوي العدد المعطى.

نحن نستخدم طريقة حساب شاملة : أوجد كل الأعداد الصحيحة الموجبة التي مربع أي منها أقل من أو يساوي العدد المعطي عند ذلك يكون أكبر هذه الأعداد هو الجذر التربيعي الصحيح

```
main( )
{
float x;
cout <<  " Enter a positive number : ";
cin >> x;
for ( int n=1 ; n*n <= x; n++)
;                                   // the null statement
cout <<  " the entegersquare root of  " << x <<  "is  "<< n-1 << endl;
```

}
Enter a positive number : 1234.56
The integer square root of 1234.56 is 35

نبدأ ب n=1 و نستمر في زيادة حتى يكون مربعn اكبر من x.x عندما تنتهـي الحلقـة n*n > x التكرارية forتكون nهي أصغر عدد صحيح مربعه أكبر من xلأذلك n-1يكون هـو الجـذر التربيعي الصحيح لـ x

لاحظ استخدام الجملة الصفرية (;) بداخل الحلقة for كل شئ كنا في حاجة إلـى عملـه فـي الحلقة تم عمله داخل أجزاء تحكم الحلقة لكن الفاصلة المنقوطة مازالت ضرورية فـي نهايـة الحلقة.

١٦. اكتب و نفذ برنامج يستخدم مباشرة عامل القسمة "/" وعامـل البـاقي % لقسـمة أرقـام صحيحة موجبة الخوارزم المستخدم هنا يطبق على الكسر n/dيكرر طرح dمـن nإلـى أن يصبح قيمة nاقل من dعند هذه النقطة تكون nهى الباقي وعدد مـرات التكـرارq اللازمـة للوصول لها يكون خارج القسمة:

```
main( )
{
int n, d, q, r;
cout <<  " Enter numerator : ";
cin >> n;
cout <<  " Enter denominator : ";
cin >> d;
for (q = 0 , r = n ; r>d ; q++) r -= d;
cout << n <<  " /  " << d <<  " =  " << q << endl;
cout << n <<  " %  " << d <<  " =  " << r << endl;
cout <<  "(  "<< q <<  " ) (" << d <<  ") + (" << r <<  " ) =  " << n << endl;
}
```

Enter numerator : 30
Enter denominator : 7
30/7 = 4
30 % 7 = 2
(4) (7) + (2) = 30

تكرار هذا التنفيذ أربع مرات : 30-7=23 , 23-7=16 , 16-7=9 , 9-7=2لهذا فـان خـارج القسمة هو 4والباقي ٢.لاحظ أن العلاقة التالية صحيحة دائما للقسمة على رقم صحيح.

(ناتج القسمة)× (المقام)+(الباقي)= البسط.

١٧.اكتب ونفذ برنامج لعكس ترتيب الأرقام في عدد صحيح موجب.

الحيلة هنا هى نزع الأرقام بواحد من الرقم المعطى ثم تجميعها في رقم صحيح أخر بترتيب عكسي

```
main( )
{
long  m,d,n=0;
cout<<"enter apositive integer:";
cin>> m;
while (m>0){
 d=m% 10;        //  d will be the right-most digit of m
m /= 10;          // then remove that digit from m
n = 10*n + d;     // and append that digit to n
}
cout <<  " The reverse is  " << n << endl;
}
Enter a positive integer : 123456
The reverse is 654321
```

هذا التنفيذ ، تبدأ m بالقيمة 123456 . فى حلقة التكرار الأولى تعطى d الرقم 6 وتقل قيمة m إلى 12,354وتزاد الى 6 . وفى الدورة الثانية . تعطى d الرقم 5 بينما تقل قيمة m الى 1234 وتزاد قيمة n الى 65 . وفى الدورة الثالثة تعطى d الرقم 4 وتقلل m الى 123 وتزاد قيمة الى 654 . ويستمر ذلك حتى الدورة السادسة حيث تعطى d الرقم ١. وتقلل m الى صفر وتزاد n الى654321.

١٨. أعد كتابة الحلقة التكرارية forمن المثال 6.3 مستخدما عامل التعبير الشرطي مكان الأمر if

التعبير الشرطي(n < min ? n : min) يصبح n إذا كانت n <min ويصبح min عدا ذلك . لذلك فان إعطاء هذه القيمة إلى min يكافئ السطر الأول في الحلقة التكرارية for بالمثال :

بالمثل إعطاء القيمة max=(n > max ? n : min)

يماثل السطر الثاني من حلقة for التكرارية الأخرى.

```
for (min  = max  = n ; n > 0 ;) {
```

```
min =(n < min ? n : min);       //min and max are the smallest
max =(n > max ? n ; min);       //and largest of the n that
cin >> n;                       //have been read so far
}
```

لاحظ فى هذا النموذج أننا لم نستخدم ما يكافئ else if

١٩. طبق الخوارزم الاقليدى لإيجاد المشترك الأعظم لرقمين صحيحين موجبين

يحول الخوارزم الاقليدى رقمين صحيحين موجبين (m,n)الى رقمين (d,o) بالقسمة المتتالية للرقم الأكبر على الرقم الصحيح الأصغر مع استبدال الأكبر بالباقي . عندما يصبح صفرا فان الرقم الأخر يصبح هو القاسم المشترك الأعظم للرقمين الأصلين (وكذلك للأزواج المتوسطة من الأرقام)

كمثال إذا كانت n =112, m=532 فان الخوارزم الاقليدى يخفض الرقمين (112,532) إلى (28,0) كالتالي .

(28,5) (84,28) (112,84) (532,12)

ولذلك فان 28 هو القاسم المشترك الأعظم بين 112,532 يمكن التحقيق من هذه النتيجة حيث ان , 112=28x4, 532=19x28 ان جدوى الخوارزم الاقليدى هنا هو كل زوج فى التتابع له نفس مجموعة القواسم التي هي بالضبط مجموعة المعاملات للقاسم المشترك الأعظم . في المثال السابق تلك المجموعة هي (28,14,7,4,2,1). إن السبب في عدم تغير هذه المجموعة من القواسم المشتركة خلال عملية الاختزال هو أنه عندما m=n.q+r فان الرقم يكون مشترك لكل من n.mاذا كان فقط هو قاسم مشترك بين r.n

```
main( )
{                               //begin scope of main ( )

int m,n,r ;
cout<<Enter two positive integers :";
cin >> m >> n;
if(m<n){ int temp=m;  m=n ;n =temp;)   //make m>=n
cout<<  " the g.c.d of "<< m <<" and  " << n << "is";
while(n>0)
r = m % n
m = n;
n = r;
}
```

```
    cout << m <<endl;
}
enter two positive integers:532    112
the g.c.d of 532 and 112 is 28
```

٢٠. أكتب واختبر برنامج يقرأ عدد من أزواج الأرقام الحقيقية (x,y) ثم يحسب المعادلة التقريبية بأقل المربعات على هيئة خط لمجموعة من النقاط . باستخدام المعادلة y=mx+b حيث

$$M= \frac{(\sum xy)\ \overline{y}(\sum x)}{(\sum xx)-\overline{x}(\sum x)}$$

$$b= \overline{y}-m\overline{x}$$

حيث $\overline{x}$ هو متوسط قيم x للنقط المعطاة.

نستعمل التدقيق المزدوج double precision للأرقام الحقيقية لتقليل الخطأ الناتج عن التقريب .

```
main( )
{
int n;                    //number of data points
double x,y,sumx =0.0 sum y=0.0,sumxx=0.0, sumxy=0.0;
cout<<" how many points: ";
cin >> n;
cout<<" enter "<< n << "pair, per line:\n";
for(int  i=l; <=n ; i++) {
cout<< ' \t ' << I << " : ";
cin>> x >>y;

sumx+=x;          //accumulate the sum of sum of the x's in sumx
sum y+=y;         // accumulate the sum of the y's in sum y
sum xx+=x*x;     //accumulate the sum of  the x*y in sum xx
sum xy += x*x ;   // accumulate the sum of the x*y in sum xy
}


double mean x=sumx/n;
double mean y= sumy/n;
double   m =(sumxy –mean y*sum x)/(sum xx-mean x*sum x);
```

```
double b=mean y-mean x*m;
cout <<  "the equation of the regression line is :\n"
<"\ty="<<m <<"x+"<<b<< endl;
}

how many points :4
enter 4 pair, one pair per line
1: 1.0 5555.04
2: 2.06666.07
3: 3.07777.05
4: 4.08888.09
the equation of the regression line is :
y=1111.01+4444.03
```

كل من عمليات الجمع الأربعة sum x .sum y .sum xx sum x y. تتراكم خلال حلقة الدخل.بعدها تحسب المتوسطات mean x ,mean y ثم تستخدم في المعادلة لحساب الميل m والتقاطع مع المحور b.y للخط التقريبي

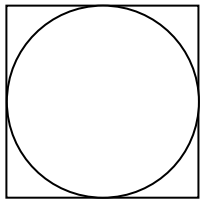
أن خرج هذا البرنامج مفيد جدا . أن الخط التقريبي يتفق إلى أقرب درجة مع البيانات المعطاة . ذلك الخط في حالة البيانات المعطاة يكون على هذا الصورة

y=1111.01x+4444.03

وهو أقرب خط ممكن بمعنى أن مجموع مربعات الفرق في اتجاهy والنقط المعطاة يكون حد أدنى .قيمة هذه النتيجة هو إمكانية استخدامها للتطابق والامتداد . كمثال لتخمين قيمةy التي تناظر قيمة x=3.2 معوضين هذه القيمة في المعادلة

y=1111.01(3.2)+4444.03=7999.26

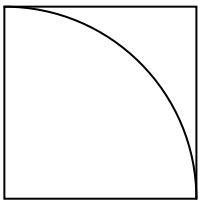
٢١.استخدام طريقة مونت كارلو لحساب النسبة التقريبية π. لقد سميت طريقة محاكاة مونت كار لو بعد الملهى فى موناكو. إنها تتكون من سحب نقاط عشوائيا وعد تلك النقاط التي تلى بشرط معين .يمكن استخدمها لحساب قيمة π بمحاكاة إلقاء السهم على لوحة دائرية مثبتة على مربع



إذا كانت الأسهم متساوية الاحتمال في الاصطدام بآي نقطة في المربع فان النسبة التي اصطدام داخل الدائرة تقترب بين النسبة بين مساحة الدائرة إلى مساحة المربع .إذا كان للمربع جوانب طولها 2.0 فان تلك النسبة تكون $(\pi r^2)/(s^2)=\pi/4$

أي أربعة أمثال تلك النسبة يقترب من π

من الأسهل استخدام ربع دائرة نصف قطرها الوحدة وتقع الوحدة في الربع الأول كما هو موضح



بهذه الطريقة فان الإحداثيات عشوائيا جميعها خلال النطاق 5.5 الى 1.5 وتكون مساحة المربع هي الوحدة ومساحة ربع الدائرة هي π/4 أى أن النسبة لا تزال π/4

```
#include<iostream.h>
#include<stdlib.h>
   #include<time.h>
main( )
  {
const long int tosses=1000 ;    //toss 1000dars
long int hits=0;
float x,y;
unsigned seed = time ( NULL);
srand(seed);
for(long int i=0; i<tosses; i++)
{
x=float(rand())/RAND-MAX
y=float (rand ())/RAND-MAX
if (x*x+y*y<1)++hits;
}
cout<<4.0 *hits/tosses<<endl;
}
3.142
3.13504
```

كلا التنفيذين يعطى تقدير النسبة π وهو مضبوط لأقرب ثلاث أرقام عشرية . يمكن تحسين الدقة بإلقاء مزيد من الأسهم ولكن على حساب الزيادة في وقت التنفيذ للبرنامج .

22 . **قلد لعبة** Monty Hall

سميت لعبة Monty Hall بعد مقدم ألعاب التليفزيون حيث يمكن للمتنافس أن يفوز بسيارة جديدة بتخمين الباب الذي تقف السيارة وراءه. أصبحت تلك اللعبة محبوبة في التسعينيات لان أحسن مبدأ للعب هو عكس التوقعات . يختار المتنافس باب واحد وعندها يفتح Monty أحد الأبواب الأخرى التي ليس خلفها السيارة وعند ذلك فان المتنافس له خيار تغيير اختياره لباب ثالث. يندهش معظم الناس ليتعلموا أن المتنافس له حظ مضاعف ليكسب السيارة إذا قام بتغيير اختياره . يمكن توضيح بواسطة الاحتمالات المشروطة ولكن لغالبية الناس يكون التمثيل بالحاسب أكثر إقناعا.

```
#include<iostream.h>
#include<stdlib>h>
include<time.h>
 main ( )
{
cout <<"this is the monty hall game.\in you see there doors "
<< "befor you .one of them has a new car behind it.\n "
<< "you will choose one of the doors. then ,befor you"
 << "get to see which\n door has the car  behind it ,monty"
<< "will give you the chance to chance \n your choice after"
<< "showing you that one of the other door has \n nothing "
<<"behind it>\n";
unsigned seed=time(NULL)
srand (seed )
 int car ,choice ,open, option;
   car=rand(  )%3+1;                  //random integer form 1 to 3
 cout << "which door do you  choose (1/2/3) : ";
cin >>choice
if(car = =1 && choice = =1)  (open =3; option=2;)


if(car = =1 && choice = =2)   (open=3; option=1;)
```

```
if (car = =1 && choice ==3)  (open=2;option=1)

if (car = =2 && choice==1)    (open=3 ;option=2)

if (car = =2 &&  choice = =2)     (open=1; option =3)

if(car = =2 && choice = = 3)      (open=1; option=2)

if(car = =3 && choice = =1)      (open=2; option=3)

if(car = =3 && choice = =2)       (open=1;option=3;)

if(car = =3 && choice = = 2)      (open=2;option=1;)

cout<< "monty shows that there is no car behind door number
"
<<open <<" .\n do you want to change your choice to door"

<<" number" << option<< " ?(y | n):;

char answer;
cin >>answer

if(answer = = 'y' || answer = = 'y') choice=option;

cout << " door number " <<car << " has the car behind it. \n ";

<< "since your final choice was door number " <<choice ;

if (choice = = car) cout << ",you won the car!\n";

else cout <<",you did not win .\n";
}
```

هذه هي لعبة Monty Hall

أنك ترى أمامك ثلاثة أبواب .إحداهم خلفه سيارة جديدة .أنك ستختار أحد الأبواب وقبل أن تتعلم أي الأبواب خلفه السيارة سيعطيك Monty الفرصة لتغير اختيارك بعد أن يعلمك أن احد الأبواب الأخرى ليس خلفه سيارة ؟أي باب ستختار (1 | 2 | 3):3.
سيظهر Monty أنه لا توجد سيارة خاف الباب رقم 1 . هل تريد تغيير اختيارك لباب رقم 2؟ (y/n):nالباب رقم 2 خلفه السيارة.
حيث أن اختيارك نهائي هو الباب رقم 3 فانك لم تكسب

23 طبق خوارزم البابليون لحساب الجذر التربيعي للرقم 2 .
خوارزم البابليون (سمي هكذا لأنه كان يستخدم بواسطة البابليون القدامى)لحساب الجذر التربيعي لـ x بالتبدل المتكرر لتقدير الأقرب 2/(x+2/x)لاحظ أن هذه ببساطة المتوسط 2\x,x

```
#include<iostream.h>
# include< math.h>          //needed for the fabs( ) function
main( )
{
  const double TOLERANCE=5e-8
double x=2.0;
while(fabs(x*x-2.0)>tolerance ) {
cout<<x<<endl;
x=(x+2.0/x)/2.0;          //average of x and 2/x
}
cout << "x= "<<x << " ,x*x "<< x*x << endl;
}
```

2
1.5
1.41667
1.41422
x=1.41421, x*x

نستخدم هنا السماح بقيمة 5e=0.0000005 لضمان دقة حتى 7خانات عشرية. الدالة () fabs (عن القيمة المطلقة للرقم الحقيقي)والمعرفة في الملف التقدمي

<math.h> تعيد القيمة المطلقة لتعبير المرسل لها .ويستمر التكرار حتى تصبح x*x فى حدود المسموح به من 2

مسائل إضافية

24.3 حول الحلقة for إلى حلقة while

```
for(int  i=20;  i>10;i--)
                    cout<<i*I;
```

25. نفذ البرنامج في المثال 13.3لإيجاد أطوال أنواع c++ الأساسية الاثنى عشر في نظامك

26 .نفذ البرنامج في المثال التالي 14.3 لإيجاد السماح ومدى المقدار للأرقام الحقيقة في نظامك

27.أوصف الخرج من الجزء التالي

```
 int f0=f1=f2=1
for (int i=0; i<10;i++ )
f0=f1
f1=f2
f2=f0=f1;
cout<< f2 << endl;}
```

28.صف الخرج الناتج عن الجزء التالي

```
for (int i=0; i<8; i++)
if (i%2= = ) cout<<i+3<<endl;
else if (i%3= =0)cout<<i+3<<endl;
else if (i%5= =0)  cout<<i*i<<endl;
else cout <<i<<endl;
```

29. صف الخرج الناتج عن الجزء الثاني

```
int i=0
while(++i<=9){
if(i==5) continue;
cout<< i<<endl;
```

30. صف الخرج الناتج عن الجزء التالي

```
int i=0 ;
while (i<5){
if(i<2){
I +=2
continue;
}
else cout <<++i<<enal;
cout<< "Bottom of loop.\n";
```

31.3 في أنواع double 64 بت يستخدم 52 بت لتخرين الرقم بينما تستخدم 11 بت لتخرين الأس.

أ-كم دقة الأرقام التي تمثلها 64 بت من النوع double ؟

ب-ما هو مدى المقدار الذي تمثله 64 من النوع double ؟

32. أكتب برنامجا ليقرأ درجة الحرارة بالتقدير المئوي ويطبع المكافئ بدرجات الفهرنهايت . كمثال إذا دخل المستخدم 75.4 درجة مئوية يكون الخرج 135.72 بالتقدير الفهرنهايتى .

33. أكتب برنامج يحول السنتيمتر إلى البوصلة . كمثال إذا دخل المستخدم 52.7 لطول بالسنتيمتر يكون الخرج 20.748 بوصة .

34 . أكتب برنامج يحول الأوزان من الرطل إلى الكيلو جرام . كمثال إذا دخل المستخدم 160 للوزن بالرطل يكون الخراج 72.5748 كجم (الرطل يساوى 0.453592 كيلو)

35. اكتب برنامج ليقرا نصف قطر كرة ويطبع مساحة سطحها وحجمها .

36.عدل ونفذ البرنامج في المثال 1.3 حتى يطبع أيضا مربع n-1 ومربع n

37. عدل البرنامج في المثال 1.3 حتى يستخدم الحلقة do ...while

38. عدل البرنامج في المثال 1.3حتى يستخدم الحلقة for

39. الكتب ونفذ برنامجا مثل الذي بالمثال 2.3 والذي يطبع المجموعة أول n تكعيب .

40. عدل البرنامج بالمثال 3.3 لتستخدم الحلقة while لحساب المضروب .

41. عدل البرنامج بالمثال 3.3 لتستخدم الحلقة for لحساب المضروب .

42. عدل البرنامج بالمثال 3.3 لتستخدم الحلقة do...while لحساب المضروب مع استخدام متغير تحكم بداخل الحلقة والتي تزيد n بدلا من إنقاصها.

43. عدل البرنامج بالمثال 3.3 لتستخدم الحلقة do ... while لحساب المضروب مستخدما متغير للتحكم داخل الحلقة والذي يزيد n بدلا من إنقاصه.

44 . عدل البرنامج بالمثال 3.3 لتستخدم الحلقة for مستخدما متغير للتحكم داخل الحلقة ليزيد n بدلا من إنقاصها .

45 . اكتب ونفذ برنامج ليقرأ عدد صحيح موجب n ويقرأ عدد ٢٠ أرقام صحيحة إضافية ويطبع مجموعها لتستخدم حلقة do ... while .

46. اكتب ونفذ برنامجا ليقرأ عدد صحيح موجب n ثم يقرأ عدد n من الأرقام الصحيحة الأخرى ثم يجمع مجموعها . استخدم الحلقة التكرار for

48. عدل البرنامج في المثال 14.3 ليطبع الدقة ومدى المقدار للنوع long double ببساطة يدل على LDBL FLT , double long- float

49. اكتب ونفذ برنامجا ليقرأ رقم صحيح موجب عندما يطبع مثلث من النجوم في نفس هذه السطور استخدام الحلقة for كمثال إذا كانت n=4 فان الخرج يكون على الشكل .

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

50. الكتب ونفذ برنامجا ليقرأ عدد صحيح موجب n عندما يطبع معين من النجوم على 2n-1 من الصفوف استخدام الحلقة for . كمثال إذا كانت n=4 فان الخرج يكون على هذا الشكل

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

51. اكتب ونفذ برنامج يستخدم مباشرة عامل القسمة / وعامل الباقي % لقسمة رقم صحيح سالب على صحيح موجب . أنظر المسألة 16.3 والمثال 16.1

52. أعد حل المسألة 19.3 مستخدما الحلقة do...while بدلا من الحلقة while .

53. عدل برنامج الجذر التربيعي للرقم الصحيح بالمسألة 15.3 حتى تنفيذ بأكثر كفاءة . استخدم خوارزم البحث الثاني بدلا من خوارزم البحث الخطى . أولا اختبر ما إذا كان الرقم الصحيح الموجب المعطى x أقل من 9 إذا لم يكن اطبع أما 0 (إذا كانت x<1) أو١ (إذا كانت x<4) أو 2 ،ثم نعود . إذا كانت xz9 فان الجذر التربيعي الصحيح لها يكون بين x/2,2 ز اقسم هذه الفترة ثم قارن n.n x حيث n هي منتصف تلك الفترة . استخدم المقارنة لتحديد مكان المنتصف الذي يحتوى على الحل . كرر هذه العملية على منتصف المسفات . استخدم الأرقام الصحيحة فقط للنقط الطرفية والمنتصف للمسافات . عندما نقطة المنتصف أكبر من النهاية اليسرى بما يساوى 1 فإنها تكون الحل .

55 .عدل البرنامج بالمثال 14.3 ليطبع الدقة ومدى المقدار للنوع double

56. عدل برنامج المعادلة التربيعية في المسألة 20.2 ليطبع المعادلة على شكل استخدامها فى الحساب .كمثال إذا كانت c=3,b=0,a=1 فان البرنامج يطبع x^2-3=0 بدلا 1x ^2+0x+-3=0

57. عدل برنامج المعادلة التربيعية في المسألة 20.2حتى تحل المعادلة فى الحالات الخاصة عند b=0,a=0 أو c=0 كمثال أنها تشير إلى أن 1.25 هو الحل للمعادلة 4x-5=0 ،وأن صفر هو حـل المعادلة 4x=0 وانه ليس هناك حل للمعادلة 5=o وأن كافة الأعداد الحقيقة هي حلـول للمعادلـة 0=0

58. اكتب واختبر برنامج ليدخل ثلاث أرقام صحيحة موجبة year, month, day ثم يطبع التـاريخ الذي يعبر عنه وعدد الأيام فى ذلك الشهر وجملة عما إذا كانت تلك السنة كبيسة . كمثـال اذا كانت الأرقام الثلاث 1997,4,6 عندئذ يطبع البرنامج april 6,1997 (بدلا من 4/6/97) شهر ابريل به 30 يوم وسنة 1997 ليست كبيسة .

59. اكتب واختبر برنامجا ليدخل أربعة أرقام صحيحة وموجبة days,yeas,month,day ثم يطبـع تاريخين. تاريخ معبر عن year,month,day والتاريخ الذى يكون بعد days

كمثال إذا كانت الأرقام الأربعة هى 6 4 ,1997, 100 حينئذ فان التاريخين المطبوعين يكونان
(4/6/97+100 days) july 15.1997<(4/6/97) april6< 1997

60. عدل برنامج التوافق الخطى (المسألة 20.3) بحيث أنه بعد حساب معادلة خط التوافق، يسمح للمستخدم بحساب إحداثيات بينية بإدخال x قيم وإخراج قيم y المناظرة والمحسوبة من المعادلة

61. عدل لعبة Monty hall (مسألة 22.3) بحيث للمستخدم اللعب بصورة متكررة فى تنفيذ واحد للبرنامج . عد عدد مرات الكسب للاعب ويطبع النسبة المئوية للفوز فى نهاية البرنامج .

62.3 عدل برنامج خوارزم البابليون (المسألة 23.3) لكي يحسب الجذر التربيعي لرقم موجب t يتم إدخاله تفاعليا . احسب متوسط التدرج x مع (x+t/x)/2

إجابات لأسئلة المراجعة

1.3 الحد الأدنى للتدرجات عندما

أ- حلقة while تعمل عدد0

ب- حلقة do...while يمكن أن تعمل عدد ١

2.3 أنها حلقة لا نهائية . متغير التحكم n لا يتغير

3.3 . ليس هناك فرق بين هاتين المقطعين إلا إذا احتوت s على خبر continue . كمثال حلقة for الثالثة ستتدرج أربع مرات ثم تنتهي كالمعتاد . ولكن الحلقة while ستكون حلقة لا نهائية .

```
for (i=0; i<4;  i++ )
if(i==2) continue;
i=0;
while( i<4) {
if(i= =2)continue;
i++ ;
}
```

4.3 الثابت لم يتم إعطاؤه قيمة ابتدائية . كل الثوابت لابد من إعطاءها قيم ابتدائية عند الإعلان عنها .

5.3 حلقة لا نهائية هي التي لا تنتهي . عامة تعتبر هذه الحلقة عادة برمجة سيئة لان البرنامج الذي يحتويها ينتهي بصورة عادية .ومع ذلك فان حلقة لا نهائية ظاهرة مثل التالية يمكن أن تكون مفيدة

```
 while ( 1){
cin>>n;
 if(n= =0) break;
```

```
process( n) ;
}
```

جملة break ستنتهي الحلقة بمجرد إدخال 0 . هذه مفيدة لأنها تسمح للخطوات بأن تكون مختصرة بعض الشيء عما إذا كانت الحالة (n= =0) ستستعمل مع جملة while

6.3جملة break يمكن استخدامها لإنهاء حلقة من خلال منتصف منطقتها . المثال السابق يوضح هذه الطريقة .

7.3 نظرا لخطا التقريب القيمة المضبوطة لرقم float أو double من المستبعد أن تكون كما نتوقع. ولذلك فان شرطا مثل if(z = = c)….. يجب تجنب استخدامه