رحم الله امرأ أهْدَى إلىَّ عُيُوبي

سيدنا عمر بن الخطاب رضي الله عنه

الجمهورية العربية السورية
جامعة دمشق
كلية الهندسة المعلوماتية

البرمجة

Engineer: khaled yassin alsheikh
المهندس خالد ياسين الشيخ

الجمهورية العربية السورية
كلية الهندسة المعلوماتية
جامعة دمشق
حل مسائل البرمجة1.

مسائل مادة البرمجة١
م.خالد الشيخ

بسم الله الرحمن الرحيم
قال تعالى " و قل ربي زدني علما "

السؤال الأول:

اكتب برنامج بلغة باسكال القياسية لإيجاد القاسم المشترك الأعظم لـــ N عدد صحيح موجب مدخل لا يساوي الصفر،
مثال: القاسم المشترك الأعظم للأعداد 90 ، 120 ، 60، 150 ، 180 هو 30.
القاسم المشترك الأعظم للأعداد 20 ، 5 هو 5. **و هكذا دواليك**
ملاحظة:يجب إدخال عددين على الأقل.
الحل:

الكود

```
program test;
const max=40;
function GCD(a,b:integer):integer;
begin
while(a<>b)do
begin
if(a>b)then
a:=a-b
else
b:=b-a;
end;
GCD:=a
end;
function GCD_N(n:integer):integer;
var i:integer; f,a:array[1..max]of integer;
begin
for i:=1 to n do
begin
write('enter the number',i,' ');
readln(a[i]);
end;
f[1]:=gcd(a[1],a[2]);
for i:=1 to n do
f[i+1]:=gcd(a[i],f[i]);
GCD_N:=f[n+1];
end;
var n,i:integer;
begin
writeln('enter the counter number>=2');
readln(n);
while(n<2)do
begin
writeln('enter number>=2');
readln(n);
end;
writeln(GCD_N(n));
readln
end.
```

السؤال الثاني :

اكتب برنامج بلغة باسكال لإيجاد المضاعف المشترك الأصغر لـــ N عدد صحيح موجب مدخل لا يساوي الصفر،

مثال: المضاعف المشترك الأصغر للأعداد 12 ، 6 ، 18 هو 72.

المضاعف المشترك الأصغر للأعداد 2 ، 8 هو 8.

المضاعف المشترك الأصغر للأعداد 9 ، 18 ، 27 هو 54 **و هكذا دواليك**

ملاحظة:يجب إدخال عددين على الأقل.

الحل:

الكود

```
program test;
const max=40;
function GCD(a,b:integer):integer;
begin
while(a<>b)do
begin
if(a>b)then
a:=a-b
else
b:=b-a;
end;
GCD:=a
end;
function LCM(a,b:integer):integer;
begin
LCM:=(a*b)  div GCD(a,b);
end;
function LCM_N(n:integer):integer;
var i:integer; f,a:array[1..max]of integer;
begin
for i:=1 to n do
begin
write('enter the number',i,' ');
readln(a[i]);
end;
f[1]:=LCM(a[1],a[2]);
for i:=1 to n do
f[i+1]:=LCM(a[i],f[i]);
LCM_N:=f[n+1];
end;
var n,i:integer;
begin
writeln('enter the counter number>=2');
readln(n);
while(n<2)do
begin
writeln('enter number>=2');
readln(n);
end;
writeln(LCM_N(n));
readln
end.
```

السؤال الثالث :

اكتب تابع بلغة باسكال و سمه square لإيجاد مربع عدد صحيح (موجب و سالب) وفق الطريقة التالية :

- مربع أي عدد صحيح (موجب أو سالب) يساوي عدد الأعداد الفردية ابتداء من العدد واحد حيث عدد الأعداد الفردية هذه يساوي عدد بمقدار العدد المراد إيجاد مربعه بالقيمة المطلقة.

مثال: مربع العدد 5 هو (1+ 3+5+7+9=25)

مربع العدد 5- (1+3+5+7+9= 25) **و هكذا دواليك...**

الحل:

الكود

```
Program test;
function square (x: integer):integer;
Var i,sum: integer;
begin
Sum:=0; i:=1;
while(x>0)do
begin
sum:=sum+i;
i:=i+2;x:=x-1;
end;
while(x<0)do
begin
sum:=sum+i;
i:=i+2;
x:=x+1;
end;
square:=sum;
end;
var n:integer; c:char;
Begin
repeat
writeln('enter the number');
readln(n);
n:=square(n);
writeln(n);
writeln('press the Y or y for exit');
readln(c);
while(not(c  in ['y','n','Y','N']))do
begin
writeln('press th y for exit or n for continue');
readln(c);
end;
until(c in ['y','Y']);
end.
```

السؤال الرابع:

اكتب تابع بلغة باسكال القياسية و سمه rotation يعمل على تدوير عدد حقيقي إلى أقرب عدد صحيح :

مثال تم إدخال العدد الحقيقي 1.5 فيكون الخرج العدد الصحيح 2.

تم إدخال العدد الحقيقي 1.4 فيكون الخرج العدد الصحيح 1.

تم إدخال العدد الحقيقي 2.49 فيكون الخرج العدد الصحيح 2.

تم إدخال العدد الحقيقي 2.54 فيكون الخرج العدد الصحيح 3. **و هكذا دواليك...**

الحل:

الكود

```
Program test;
function rotation(num:real):integer;
var temp:real; res:integer;
begin
temp:=num;
while(temp>1)do
temp:=temp-1;
if(temp<0.5)then
res:=trunc(num-temp)
else
res:=trunc((temp-1)*(-1)+num);
rotation:=res;
end;
var n:real;
begin
readln(n);
writeln(rotation(n));
readln
end.
```

السؤال الخامس:

اكتب بلغة باسكال القياسية:

- تابع intercalary يتحقق من كون سنة ما كبيسة أم لا علما أن السنة الكبيسة تقبل القسمة على العدد 4 و 400 معاً أما في حال أن السنة تقبل القسمة على 4 و لا تقبل القسمة على العدد 400 و تقبل القسمة على العدد 100 فهذا يعني أنها ليست كبيسة و في حال أن السنة تقبل القسمة على 4 و لا تقبل على 400 و لا تقبل القسمة على العدد 100 فهذا يعني أنها سنة كبيسة. حيث البرنامج يقوم بطباعة الكلمة 'ok' في حال كانت السنة كبيسة.
- تابع number يرد عدد السنوات بين سنتين مدخلتين مثلا بين 1992 و 2000 عدد من السنوات هو 7 سنوات.

الكود

```
Program test;
function intercalary(year:word):boolean;
var ok:boolean;
begin
ok:=false;
if(year mod 4=0)then
if(year mod 400=0)then
ok:=true
else
if(year mod 100=0)then
ok:=false
else
ok:=true
else
```

```
ok:=false;
intercalary:=ok;
end;
function number(year1,year2:word):word;
var i,res:integer;
begin
res:=0;
for i:=year1+1 to year2-1 do
res:=res+1;
number:=res
end;
var
y,y1,y2:word;
begin
readln(y);
if(intercalary(y))then
writeln('ok');
readln(y1,y2);
if(y1>y2)then
begin
y1:=y1 xor y2;
y2:=y1 xor y2;
y1:=y1 xor y2;
end;
writeln(number(y1,y2));
readln
end.
```

السؤال السادس:

أكتب بلغة الخوارزميات مع الشرح من حيث التحليل و تعقيد(خوارزمية البحث الثنائي):

- **الطريقة:**

تُعدَّ خوارزمية البحث الثنائي (Binary Search) من خوارزميات البحث الجيدة، لأنها تشترط ترتيب القائمة قبل إجراء عملية البحث. وهذا بحد ذاته سوف يسهل عملية البحث عن العنصر المطلوب. تعتمد هذه الخوارزمية على تقسيم القائمة إلى نصفين متساويين، فإذا كان العنصر المطلوب ضمن النصف الأول فيُستبعد البحث في النصف الثاني. أما إذا كان في النصف الثاني فلا يتم البحث عنه في النصف الأول. وبهذا سنقلل عمليات المقارنة إلى النصف في كل مرحلة. وتجدر الإشارة هنا إلى أن استخدام هذه الخوارزمية مع القوائم الكبيرة يعطي كفاءة أعلى فيما إذا تم استخدامها مع القوائم الصغيرة.

أما تحديد منتصف القائمة فيمكن إيجاده من العلاقة الآتية:

$$Mid = \frac{(low + high)}{2}$$

الخوارزمية:

```
Binary -Search[A, n, x]
low ← 1
high ← n
while low ≤ high do
    mid ← ( low + low ) / 2
    if A[ mid ] = x then
        return mid
    else if A[ mid ] < x then
            low ← mid + 1
        else high ← mid − 1
return "x not found"
```

- **التحليل ودرجة التعقيد:**

• **أسوأ حالة**:عندما يكون العنصر في نهاية القائمة أو أنه غير موجود فيها. وبما أن عمليات المقارنة تنفذ على جميع العناصر فالوقت اللازم سيكون :

$$T(n) = O(n \log n)$$

• **أحسن حالة**:عندما يكون العنصر في منتصف القائمة، أي في نفس موقع التقسيم (mid) وهي:

$$T(n) = 1 = O(1)$$

• **الحالة المتوسطة**: عندما تكون هناك احتمالية وجود العنصر في أي جزء من القائمة. وعليه سيكون وقت التنفيذ:

$$T(n) = \sum_{i=1}^{n} (i \times \frac{1}{n}) = \frac{1}{n} \times \sum_{i=1}^{n} i$$

$$= \frac{1}{n} \times \frac{n(n+1)}{2} = \frac{n+1}{2}$$

معلومة لطيفة :

يوجد نوعان من الأخطاء :أخطاء أثناء الترجمة compilation errors مثل استخدام متحول دون تعريفه أو كتابة عبارة بطريقة خاطئة .فهذه الأخطاء لايمكن تجاوزها، وتحول دون ترجمة وربط البرامج ويقوم المترجم بالتنبيه عن وجودها في نافذة Message.
أما النوع الثاني من الأخطاء فهي الاعتراضات التي تحدث بعد تنفيذ البرنامج، مثل إذا طلب من البرنامج إدخال رقمين لإجراء القسمة عليهما و قام المستخدم بإدخال صفر في المقسوم عليه ففي هذه الحالة يحدث الخطأ المعروف بالقسمة على صفر Division by zero أو محاولة فتح ملف غير موجود مثل هذه الأخطاء تسمى أخطاء في وقت التنفيذ run time error و ليس من السهولة أن يتوقعها المترجم لذلك فمعالجة مثل هذه الأخطاء تركت على عاتق المبرمج.

رحم الله امرأ أهْدَى إلىَّ عُيُوبي

سيدنا عمر بن الخطاب رضي الله عنه

اللهم صلي و سلم على سيدنا محمد و على آله و صحبه أجمعين

(و ما توفيقي إلا بالله)