

SECTION : SCIENCES DE L'INFORMATIQUE

EPREUVE : ALGORITHMIQUE ET PROGRAMMATION

DURÉE : 1 H

NOMBRE DE PAGES : 2

COEF : 3

Nb : L'examen comporte deux pages

Exercice N°1 : (4 points)

Soit le l'algorithme suivant :

0. Début voyelle_consonne_autres
1. Ecrire("Saisir une chaine : ")
Lire (ch)
2. $V \leftarrow 0$, $c \leftarrow 0$, autres $\leftarrow 0$
3. **Proc compter (ch, v, c, autres)**
4. **Ecrire** (« v= », v, « c= », c, « autres= », autres)
5. Fin voyelle_consonne_autres

Ecrire un algorithme récursif du module **compter**, permettant de calculer le nombre de voyelles, le nombre de consonnes et le nombre des autres caractères non alphabétiques contenus dans une chaine de caractère **ch donnée**. (Pour simplifier on ignore les caractères accentués)

Exercice N°2 : (4 points)

Une matrice carrée, de dimension impaire, est dite magique si :

la somme de chaque ligne = la somme de chaque colonne = la somme de chaque diagonale.

Exemple, pour n=3, Cette Matrice est magique, car toutes les sommes sont égales à 15.

4	9	2
3	5	7
8	1	6

Ecrire un algorithme d'un module qui vérifie si une matrice donnée est magique

Exercice N°3 : (6 points)

Soit la suite U définie de la façon suivante :

$$\begin{aligned}U_0 &= 1 \\U_1 &= 2 \\U_n &= \frac{U_{n-1} + \frac{2}{U_{n-2}}}{2}\end{aligned}$$

- Ecrire un algorithme itératif permettant de calculer et d'afficher les n premier termes de cette suite et leur somme. (Sans utiliser de tableau)
- Ecrire un algorithme récursif permettant de calculer un terme quelconque de cette suite.

Exercice N°4 : (6 points)

On se demande s'il existe une fonction générale et unique qui calcule d'une façon récursive la somme, le produit et l'élevation à la puissance, de deux nombres a et b ; Une telle fonction existe, c'est la fonction **d'Ackermann**. Notée **Ack**, et définie de la manière suivante :

$$\begin{aligned}Ack(0, a, b) &= a + 1 \\Ack(1, a, 0) &= a \\Ack(2, a, 0) &= 0 \\Ack(3, a, 0) &= 1 \\Ack(n, a, 0) &= 2 \text{ pour } n > 3 \\Ack(n, a, b) &= Ack(n-1, Ack(n, a, b-1), a)\end{aligned}$$

- Ecrire une fonction récursive permettant de calculer **Ack(n,a,b)**, avec n, a et b trois entiers données.
- Ecrire une procédure permettant de varier n de 1 à 3 et a et b de 0 à 4, en affichant à chaque fois le résultat de la fonction précédente pour vérifier que :

$$\begin{aligned}Ack(1, a, b) &= a + b \\Ack(2, a, b) &= a \times b \\Ack(3, a, b) &= a^b\end{aligned}$$

Bon travail !

{Correction Exercice 1 devoir de controle DC2-1}

```

Program voyelles_consonnes_autres ;
uses wincrt;
var
ch:string;

procedure compter(ch:string; nbv,nbc,autre:byte);

begin
ifch='' then begin
writeln('nombre de voyelle =',nbv);
writeln('nombre de consonne=', nbc);
writeln('nombre des autres=', autre);
end

else if upcase (ch[1]) in ['A','E','I','O','U','W','Y'] then
compter(copy(ch,2, length(ch)-1),nbv+1,nbc,autre)
else if upcase (ch[1]) in
['B','C','D','F','G','H','J','K','L','M','N','P','Q','R','S','T','V','X','Z'
]
thencompter(copy(ch,2, length(ch)-1),nbv,nbc+1,autre)
else compter(copy(ch,2, length(ch)-1),nbv,nbc,autre+1);

end;

begin
write('Saisir une chaine:');
readln(ch);

compter (ch,0,0,0);
end.

```

{Correction Exercice 2 devoir de controle DC2-1}

```

programmatrice_magique;
useswincrt;
const
nmax=20;
type
tmat=array[1..nmax,1..nmax] of byte;
var
mat:tmat;
n:byte;
procedurelecture_n(var n:byte);
var
s:string;
pe:integer;
begin
repeat
write('n= ');
readln(s);
val(s,n,pe);
until (n>=2) and (pe=0);
end;

procedureremplir (var m:tmat; n:byte);
var
i,j:byte;
begin
for i:=1 to n do
begin

```

```

writeln('Ligne ',i);
for j:=1 to n do
begin
readln(m[i,j]);
end;
end;

end;

{_____}
fonctionsomme_ligne(l:byte):integer;
var
s,c:integer;
begin
s:=0;
for c:=1 to n do s:=s+mat[l,c];

somme_ligne:=s;
end;

fonctionsomme_colonne(c:byte):integer;
var
s,l:integer;
begin
s:=0;
for l:=1 to n do s:=s+mat[l,c];

somme_colonne:=s;
end;

function somme_diag1(n:byte):integer;
var
s,i:integer;
begin
s:=0;
for i:=1 to n do
s:=s+mat[i,i];

somme_diag1:=s;
end;

function somme_diag2(n:byte):integer;
var
s,i,j:integer;
begin
s:=0;
j:=0;
for i:=n downto 1 do
begin
j:=j+1;
s:=s+mat[i,j];
end;

somme_diag2:=s;
end;

{_____}

```

(***** Le travail demandé *****)

```

functionest_magique(mat:tmat;n:byte):boolean;
var
l,c:byte;
s1:integer;
ok:boolean;
begin

        s1:=somme_ligne(1);
l:=1;
repeat
l:=l+1;
until (somme_ligne(l)<>s1) or (l=n);

ok:= (somme_ligne(l)= s1) ;

if ok then begin

c:=1;
repeat
c:=c+1;
until (somme_colonne(c)<>s1) or (c=n);

ok:= somme_colonne(c)=s1;

if ok then begin
ok := (somme_diag1(n)=s1) and (somme_diag2(n)=s1);
end;
end;

est_magique:=ok;
end;

(*****

procedureafficher(mat:tmat;n:byte);
varl,c:byte;
begin

writeln;
writeln;
for l:=1 to n do
begin
for c:=1 to n do
write(mat[l,c]:4);
writeln;
end;

writeln;
writeln;

        if est_magique(mat,n) thenwriteln('Cette matrice est magique!')
elsewriteln('Cette matrice n''est pas magique!');

end;

BEGIN
repeat
        lecture_n(n);
remplir(mat,n);
afficher(mat,n);

```

```
until 'le rouge'='le noir';
END.
```

{Correction Exercice 3 a) devoir de contrôle DC2-1}

```
program suite_exercice3_a;
useswincrt;
var n:integer;

procedure lecture (var n:integer);
begin
repeat
write('N=');
readln(n);
until n>=2;
end;

(***** Le travail demandé dans le devoir*****)
procedurecalculer_afficher(n:integer);
var i:integer; un,un_1,un_2,s:real;
begin
un_2:=1; write(1:6);
un_1:=2; write(2:6);

s:=1+2;
for i:=2 to n-1 do
begin
un:=(un_1 + 2/un_2)/2;
write(un:8:2);

s:=s+un;
un_2:= un_1;
un_1:= un
end;

writeln;
writeln('Somme=', s:0:2);
end;
(*****)

begin
lecture(n);
calculer_afficher(n);

end.
```

{Correction Exercice 3 b) devoir de controle DC2-1}

```
program suite_exercice3_b;
useswincrt;
var n:integer;

procedure lecture (var n:integer);
begin
repeat
write('N=');
readln(n);
until n>=2;
end;

(***** Le travail demandé dans le devoir*****)
```

```

functioncalcul_terme(n:integer):real;
begin
if n=0 then calcul_terme:=1
else if n=1 then calcul_terme :=2
elsecalcul_terme :=( calcul_terme(n-1) +2/ calcul_terme(n-2))/2;
end;
(*****)

begin
lecture(n);
writeln('Le ',n,'ième terme= U(',n-1,')=', calcul_terme(n-1) :0:2);

end.

```

{Correction Exercice 4 devoir de controle DC2-1}

```

programsuite_ackermann;
useswincrt;

```

```

functionack(n,a,b:integer):integer;
begin
if n=0 then ack:=a+1
else if b=0 then
if n=1 then ack:=a
else if n=2 then ack:=0
else if n=3 then ack:=1
elseack:= 2
elseack:=ack(n-1, ack(n,a,b-1), a)
end;

```

```

functionpuiss(a,b:integer):integer;
begin
if b=0 then puiss:=1
elsepuiss:= a* puiss(a,b-1);
end;

```

```

procedureafficher (a,b:integer);
var i:integer;
begin
for i:= 1 to 3 do
begin
for a :=0 to a do
for b:=0 to b do
begin
write('ack(',i,',',a,',',b,')=', ack(i,a,b), '
vérification: ');
case i of
1:writeln(a,'+',b, '= ',a+b);
2:writeln(a,'*',b, '= ',a*b);
3:writeln(a,' puiss ',b, '= ',puiss(a,b));
end;
end;

writeln(#13,' Entrer pour continuer...');
readln;
end;
end;

```

```
begin  
afficher(4,4);  
end.
```