

NB : la manipulation des téléphones est formellement interdite.

Exercice 1 : (5 points)

Soit l'algorithme suivant :

```

Algorithme exo
variables i,n:entier
v:tableau[1..50] d'entier
début
  lire (n)
  pour i ← 1 à n faire
    lire(v[i])
  fin pour
  i← 1
  tantque (i<=n) et (v[i]>0) faire
    i ← i+1
  fin_tantque
  si i>n alors écrire('oui')
  sinon écrire('non')
  fin_si
fin
    
```

- 1) Traduire l'algorithme en Pascal
- 2) Dérouler l'algorithme pour n=5 et

V=

5	4	-2	0	1
---	---	----	---	---

- 3) Que fait cet algorithme ?

Exercice 2 : (4 points)

Soit un tableau T de N éléments entiers.

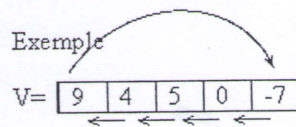
Ecrire un programme Pascal qui calcule et affiche la somme des éléments du rang (ou indice) pair et le produit des éléments du rang impair.

Exercice 3 : (6 points)

Les séquences d'instructions suivantes réalisent le décalage de manière circulaire à gauche les éléments d'un vecteur V de m composantes.

```

Z:= V[1];
for j:=1 to m-1 do
  V[j]:= V[j+1];
V[m] := Z;
    
```



Le tableau devient

V=

4	5	0	-7	9
---	---	---	----	---

Où Z est une variable intermédiaire.

Questions :

Ecrire un programme Pascal qui :

- 1- Lit les éléments de type réel de la matrice Mat de n lignes et m colonnes.
- 2- Décale de manière circulaire à gauche chaque ligne de la matrice Mat.
- 3- Affiche la matrice résultat.

Exemple

5	4	5
2	7	1
4	0	3
..
..

Devient →

4	5	9
7	1	2
0	3	4
..
..

Exercice 4 : (5 points)

Soit la fonction booléenne, appelée PARFAIT, pour vérifier si le nombre (X) est parfait ou non parfait. Sachant qu'un nombre entier positif (X) est parfait s'il est égal à la somme de ses diviseurs (*sauf lui-même*). La fonction renvoi la valeur booléenne **true** si X est parfait ou la valeur **false** si X est non parfait.

```

Function PARFAIT(X :integer) :boolean ;
Var i,S :integer ;
begin
  S := 0 ;
  for i:=1 to (X div 2) do
    if (X mod i ) = 0 then
      S := S+i;
    if S=X then
      Parfait :=true
    else
      Parfait := false;
  end;
end;
    
```

Questions :

- 1) En utilisant la fonction PARFAIT, écrire un programme PASCAL affichant tous les nombres parfaits strictement inférieurs à 1000.
- 2) Réécrire le programme en transformant la fonction PARFAIT en procédure PARFAIT.

Exemples :

Les nombres 6 et 28 sont des nombres parfaits car :

$$6 = 1 + 2 + 3$$

$$28 = 1 + 2 + 4 + 7 + 14$$

Exercice 1 : (5 points)

```

1)
program exo1;
uses winct;
var V : array[1..10] of integer;(0.5 point)
i, n : integer;
BEGIN
read(n);
for i:= 1 to n do (0.5 point)
read(V[i]);
i:=1;
while (i<n) and (V[i]>0) do(1 point)
i:= i+1;
if i>n then(1 point)
write ('oui')
else write ('non');
END.
2)

```

Instructions	n	i	V	Affichage
Lire (n)	5			
Pour i=1 lire(V[1])		1	(5, , , ,)	(0.5 point)
Pour i=2 lire(V[2])		2	(5, 4, , ,)	
Pour i=3 lire(V[3])		3	(5, 4, -2, ,)	
Pour i=4 lire(V[4])		4	(5, 4, -2, 0,)	
Pour i=5 lire(V[5])		5	(5, 4, -2, 0, 1)	
i=1		1		
TANT QUE				
(i<n) et (V[i]>0)=vrai		2		(0.5 point)
i ← i + 1				
(i<n) et (V[i]>0)=vrai		3		
i ← i + 1				
(i<n) et (V[i]>0)=faux				
SI				
(i>n)=faux				
écrire ('Non')				Non

3) L' algorithme vérifie si toutes les valeurs d' un vecteur sont strictement positives ou non. (0.5 point)

Exercice 2 : (4 points)

```

program exo1;
uses winct;
var V : array[1..10] of integer; (0.5 point)
i, n, som, prod : integer;
BEGIN
read(n);
for i:= 1 to n do (0.5 point)
read(V[i]);
som:=0;
prod:=1; (0.5 point)
for i:= 1 to n do
if i mod 2 = 0 then(1 point)
som:=som +V[i] (0.5 point)
else
prod:=prod * V[i]; (0.5 point)
write(som, prod); (0.5 point)
END.

```

Exercice 3 : (6 points)

```

Program Exo3;
Uses winct;
Var i,j,n,m: integer; Z:real;
Mat : array[1..10,1..10]of real; (0.5 point)
begin
read(n,m);
for i:=1 to n do (0.5 point)
for j:=1 to m do
read(mat[i, j]);
for i:=1 to n do (1 point)
begin
Z:= mat[i,1]; (1 point)
for j:=1 to m-1 do
mat[i,j]:= mat[i,j+1]; (1 point)
mat[i,m] := Z; (1 point)
end;
for i:=1 to n do (1 point)
begin
for j:=1 to m do
write(mat[i ,j]:6:2);
writeln;
end;
End.

```

Exercice 4 : (5 points)

```

1)
Program NombreParfait ;
uses winct;
Var N : integer ; k:boolean;
Function PARFAIT(X : integer) :boolean ; (0.5 point)
Var i,S :integer ;
begin
S := 0 ;
for i:=1 to (X div 2) do
if X mod i = 0 then
S := S+i;
if S=X then
Parfait := true (0.5 point)
else
Parfait := false;
end;
BEGIN
for N:=1 to 999 do (0.5 point)
begin
k:=PARFAIT( N ); (0.5 point)
if (k=True) then (0.5 point)
writeln(N,' est parfait');
end ;
END.

2)
Program NombreParfait ;
uses winct;
Var N : integer ; k:boolean;
Procedure PARFAIT(X : integer ; Var Z : boolean); (1 point)
Var i,S :integer ;
begin
S := 0 ;
for i:=1 to (X div 2) do
if X mod i = 0 then
S := S+i;
if S=X then
Z := true (0.5 point)
else
Z := false;
end;
BEGIN
for N:=1 to 999 do
begin
PARFAIT( N, k ); (1 point)
if (k=True) then
writeln(N,' est parfait');
end ;
END.

```