

## Exercice 1 :

- 1) Ecrire un programme qui permet d'afficher les nombres compris entre 0 et 100 en utilisant la boucle While ?

### Algorithme

Tant que	Pour	répéter
<p><b>Algorithme exo1</b>  <b>Var</b> i : entier  <b>Debut</b>            i ← 0            <b>Tq</b> i ≤ 100 <b>Faire</b>                Ecrire (i)                i ← i+1            <b>Fintq</b>  <b>Fin</b></p>	<p><b>Algorithme exo1</b>  <b>Var</b> i : entier  <b>Debut</b>            <b>Pour</b> i de 0 à 100 <b>Faire</b>                Ecrire (i)            <b>Finpour</b>  <b>Fin</b></p>	<p><b>Algorithme exo1</b>  <b>Var</b> i : entier  <b>Debut</b>            i ← 0            <b>Répéter</b>                Ecrire (i)                i ← i+1            <b>Jusqu' à</b> i=100  <b>Fin</b></p>

- 2) Modifier ce programme pour qu'il soit affiché seulement les nombres pairs ?

Tant que	Pour	répéter
<p><b>Algorithme exo1</b>  <b>Var</b> i : entier  <b>Debut</b>            i ← 0            <b>TQ</b> i ≤ 100 <b>Faire</b>                <b>Si</b> i mod 2 = 0 <b>Alors</b>                    Ecrire (i)                <b>Finsi</b>                i ← i+1            <b>FinTQ</b>  <b>Fin</b></p>	<p><b>Algorithme exo1</b>  <b>Var</b> i : entier  <b>Debut</b>            <b>Pour</b> i de 0 à 100 <b>Faire</b>                <b>Si</b> i mod 2 = 0 <b>Alors</b>                    Ecrire (i)                <b>Finsi</b>            <b>Finpour</b>  <b>Fin</b></p>	<p><b>Algorithme exo1</b>  <b>Var</b> i : entier  <b>Debut</b>            i ← 0            <b>Repeter</b>                <b>Si</b> i mod 2 = 0 <b>Alors</b>                    Ecrire (i)                <b>Finsi</b>                i ← i+1            <b>Jusqu' à</b> i=101  <b>Fin</b></p>

**Remarque :** TQ : tantque ; FTQ : fin tantque

# Programme

1)

```
File Edit Search
[ ]
program exo1_while;
var i:integer;
begin
  i:=0;
  while i<=100 do
  begin
    writeln (i);
    i:=i+1;
  end ;
readln;
end.
```

3:44

F1 Help F2 Save F3

```
File Edit Search
[ ]
program exo1_for;
var i:integer;
begin
  for i:=0 to 100 do
    writeln(i);
  readln;
end.
```

4:7

F1 Help F2 Save F3

```
File Edit Search
[ ]
program exo1_repeat;
var i:integer;
begin
  i:=0;
  repeat
    writeln (i);
    i:=i+1;
  until i=101;
readln;
end.
```

1:20

F1 Help F2 Save F3

2)

```
File Edit Search
[ ]
program exo1_while;
var i:integer;
begin
  i:=0;
  while i<=100 do
  begin
    if i mod 2=0 then
      writeln (i);
    i:=i+1;
  end ;
readln;
end.
```

8:31

F1 Help F2 Save F3

```
File Edit Search
[ ]
program exo1_for;
var i:integer;
begin
  for i:=0 to 100 do
    if i mod 2=0 then
      writeln(i);
  readln;
end.
```

8:29

F1 Help F2 Save F3

```
File Edit Search
[ ]
program exo1_repeat;
var i:integer;
begin
  i:=0;
  repeat
    if i mod 2=0 then
      writeln (i);
    i:=i+1;
  until i=101;
readln;
end.
```

10:21

F1 Help F2 Save F3

## Exercice 2 : (utiliser la boucle FOR)

Ecrire un programme qui calcule le factoriel d'un nombre entier entrée par l'utilisateur, puis vérifier le programme pour les nombres 0,4,7,8,9. Que remarquez-vous ? Proposer une solution pour pallier le problème raconté.

### Algorithme

**Algorithme** exo2

**Var** i, n, f : entier

**Debut**

Ecrire ('entrez un nombre')

Lire (n)

**Pour** i de 1 à n **Faire**

f := f\*i

**Finpour**

ecrire ('le factoriel de ', n, ' est ', f)

**Fin**

### Programme

```
program exo2;
var i,n,f:integer;
begin
writeln('Entrez un nombre');
readln(n);
f:=1;
  for i:=1 to n do
    f:=f*i;
    writeln('le factoriel de ', n, ' est : ', f);
readln;
end.
```

9:23

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make

### Exercice 3 : (utiliser une boucle de votre choix)

Ecrire un programme pascal qui Calcule le nombre des nombres pairs et impairs compris entre deux variables A et B entrées par l'utilisateur (Ex : A=1, B= 9 le nb de pair = 4, nb impaire =5).

## Algorithme

**Algorithme** exo3

**Var** i , a, b, npair, nimpair : entier

**Debut**

Ecrire ('entrez un nombre')

Lire (a)

Ecrire ('entrez un nombre')

Lire (b)

**Si** i a>b **Alors**

c←a

a←b

b←c

**Finsi**

npair←0

nimpair←0

**Pour** i de a à b **Faire**

**Si** i mod 2= 0 **Alors**

npair←npair+1

**Sinon**

Nimpair←nimpair+1

**Finsi**

**Finpour**

Ecrire ('le nombre des pairs est :', npair)

Ecrire ('le nombre des impairs est :', nimpair)

**Fin**

## Programme

```
File Edit Search Run Compile Debug Tool
=[ ] EX01_3.PAS =
program exo3;
var a,b,c,i,npair,nimpair:integer;
begin
writeln('entrez un nombre');
readln(a);
writeln('entrez un nombre');
readln(b);
if a>b then
begin
c:=a;
a:=b;
b:=c;
end;
npair:= 0;
nimpair:=0;
for i:=a to b do
if i mod 2 =0 then
npair:=npair+1
else
nimpair:=nimpair+1;
writeln('le nombre des pairs est : ', npair);
writeln('le nombre des impairs est : ', nimpair);
readln;
end.
1:1
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make
```

## Exercice 4 :

Ecrire un programme qui calcule la somme des nombres compris entre deux valeurs A et B entrées par l'utilisateur. Modifier ce programme pour calculer la somme carrée entre A et B.

### Algorithme

<p><b>Algorithme</b> exo4</p> <p><b>Var</b> i , a, b, s : entier</p> <p><b>Debut</b></p> <p>Ecrire ('entrez un nombre')</p> <p>Lire (a)</p> <p>Ecrire ('entrez un nombre')</p> <p>Lire (b)</p> <p><b>Si</b> i a&gt;b <b>Alors</b></p> <p>    c ← a</p> <p>    a ← b</p> <p>    b ← c</p> <p><b>Finsi</b></p> <p>S ← 0</p> <p><b>Pour</b> i de a à b <b>Faire</b></p> <p>    s := s + i</p> <p><b>Finpour</b></p> <p>    ecrire ('la somme est : ', s )</p> <p><b>Fin</b></p>	<p><b>Algorithme</b> exo4</p> <p><b>Var</b> i , a, b, s : entier</p> <p><b>Debut</b></p> <p>Ecrire ('entrez un nombre')</p> <p>Lire (a)</p> <p>Ecrire ('entrez un nombre')</p> <p>Lire (b)</p> <p><b>Si</b> i a&gt;b <b>Alors</b></p> <p>    c ← a</p> <p>    a ← b</p> <p>    b ← c</p> <p><b>Finsi</b></p> <p>S ← 0</p> <p><b>Pour</b> i de a à b <b>Faire</b></p> <p>    s := s +sqr( i)</p> <p><b>Finpour</b></p> <p>    ecrire ('la somme est : ', s )</p> <p><b>Fin</b></p>
--	---

## Programme

```
File Edit Search Run Comp
[ ]
program exo4;
var a,b,c,i,s:integer;
begin
writeln('entrez un nombre');
readln(a);
writeln('entrez un nombre');
readln(b);
if a>b then
begin
c:=a;
a:=b;
b:=c;
end;
s:= 0;
for i:=a to b do
s:=s+i;
writeln('la somme est : ',s);
readln;
end.
18:1
F1 Help F2 Save F3 Open Alt
```

```
File Edit Search Run Comp
[ ]
program exo4;
var a,b,c,i,s:integer;
begin
writeln('entrez un nombre');
readln(a);
writeln('entrez un nombre');
readln(b);
if a>b then
begin
c:=a;
a:=b;
b:=c;
end;
s:= 0;
for i:=a to b do
s:=s+sqr(i);
writeln('la somme est : ',s);
readln;
end.
16:16
F1 Help F2 Save F3 Open Alt
```

## Exercice 5 :

Calculer le Nième terme UN de la suite de FIBONACCI qui est donnée par la relation de récurrence :  $U_1=1, U_2=1, U_N = U_{N-1} + U_{N-2}$  (pour  $N>2$ ). N est donné par l'utilisateur.

## Algorithme

**Algorithme** exo5

**Var** i, n, u1, u2, un : entier

**Debut**

Ecrire ('entrez un nombre')

Lire (n)

$u1 \leftarrow 1$

$u2 \leftarrow 1$

**Pour** i de 3 à n **Faire**

$un \leftarrow u1 + u2$

$u1 \leftarrow u2$

$u2 \leftarrow un$

ecrire ('u', i, ' = ', un)

**Finpour**

**Fin**

## Programme

```
File Edit Search Run Compile
[ ]
program exo5;
var i,n, u1,u2,un:integer;
begin
  writeln('entrez un nombre');
  readln(n);
  u1:=1;
  writeln('u1 = ',u1);
  u2:=1;
  writeln('u2 = ',u2);
  for i:=3 to n do
  begin
    un:=u1+u2;
    u1:=u2;
    u2:=un;
    writeln('u',i,' = ',un);
  end;
  readln;
end.
```

\* 18:76

F1 Help F2 Save F3 Open Alt+F

## Exercice 6 : (utiliser la boucle Repeat)

Ecrire un programme permettant de calculer le PGCD de deux nombres en utilisant l'astuce suivante : Soustraire le plus petit des deux entiers du plus grand jusqu'à ce qu'ils soient égaux.

### Algorithme

**Algorithme** exo6

**Var** i, a, b, x, y, z : entier

**Debut**

Ecrire ('entrez un nombre')

Lire (a)

Ecrire ('entrez un nombre')

Lire (b)

**Si**  $a > b$  **alors**

$x \leftarrow a$

$y \leftarrow b$

**Si non**

$x \leftarrow b$

$y \leftarrow a$

**Finsi**

**repeat**

$z \leftarrow x - y$

**Si**  $y > z$  **alors**

$x \leftarrow y$

$y \leftarrow z$

**Si non**

$x \leftarrow z$

$y \leftarrow y$

**Finsi**

**Jusau'a**  $z = 0$

    ecrire ('le PGCD de ', a, 'et', b, ' est : ', x)

**Fin**

# Programme

```
File Edit Search Run Compile
[ ]
program exo6;
var i,a,b,x,y,z:integer;
begin
  writeln('entrez un nombre');
  readln(a);
  writeln('entrez un nombre');
  readln(b);
  if a>b then
    begin
      x:=a;
      y:=b;
    end
  else
    begin
      x:=b;
      y:=a;
    end ;
  repeat
    z:=x-y;
    if y>z then
      begin
        x:=y;
        y:=z;
      end
    else
      begin
        x:=z;
        y:=y;
      end
    until z=0;
  writeln('le PGCD de ',a, ' et ', b, ' est : ', x);

  readln;
end.
```

1:13

```
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make
```

## Exercice 7 :

En considérant la somme  $S_n$  définie pour tout entier  $n$  par :

$$S_n = 4 \sum_{k=0}^n \frac{k}{2k+1}$$

Écrire un programme qui calcule  $S_n$ .  $N$  est un entier saisi par l'utilisateur.

## Algorithme

**Algorithme** exo7

**Var**  $i, n$  : entier

$s$  : réel

**Debut**

Ecrire ('entrez un nombre')

Lire ( $n$ )

$s \leftarrow 0$

**Pour**  $k$  de 0 à  $n$  **faire**

$s \leftarrow s + 4 * (k / (2 * k + 1))$

**Finpour**

ecrire ('la somme est : ',  $s$ )

**Fin**

## Programme

```
File Edit Search Run Compile
[ ]
program exo7;
var k,n:integer;
    s:real;
begin
writeln('entrez un nombre');
readln(n);
s:= 0;
for k:=0 to n do
    s:=s+4*(k/(2*k+1));

writeln(' la somme est : ',s:3:5);
readln;
end.
```

11:32

F1 Help F2 Save F3 Open Alt+F9

## Exercice 8

Ecrire un programme pascal qui permet de convertir une phrase écrite en minuscule vers la même phrase en majuscule (utiliser les fonctions *upCase* et *length*)

### Algorithme

#### Algorithme exo8

**Var** p : chaîne de caractère

i : entier

#### Debut

Ecrire ('entrez une phrase')

Lire (p)

**Pour** i de 1 à length(p) **faire**

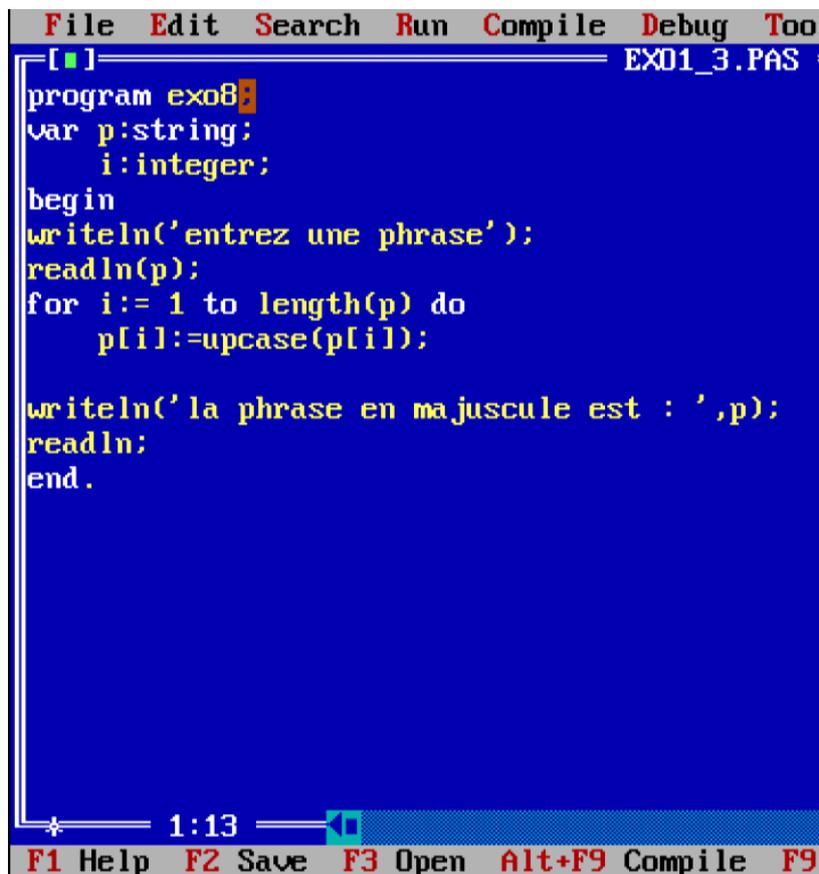
P[i] ← upcase(p[i])

**Finpour**

ecrire ('la phrase en majuscule est : ', p)

#### Fin

### Programme



```
File Edit Search Run Compile Debug Too
[ ] EXO1_3.PAS
program exo8;
var p:string;
    i:integer;
begin
writeln('entrez une phrase');
readln(p);
for i:= 1 to length(p) do
    p[i]:=upcase(p[i]);

writeln('la phrase en majuscule est : ',p);
readln;
end.
```

1:13

F1 Help F2 Save F3 Open Alt+F9 Compile F9

## Exercice 9

On veut écrire un programme Pascal permettant de coder un message selon le procédé suivant :  
permuter chaque caractère d'indice pair avec le caractère qui le précède.

Exemple : Le codage de la chaîne de caractère : "Baccalauréat" donne "aBcclauaérta"

### Algorithme

#### **Algorithme** exo9

**Var** p : chaîne de caractères

c : caractère

i : entier

#### **Debut**

Ecrire ('entrez une phrase')

Lire (p)

$i \leftarrow 0$

**TQ**  $i < \text{length}(p)$  **faire**

c  $\leftarrow$  P[i]

p[i]  $\leftarrow$  (p[i+1])

p[i+1]  $\leftarrow$  c

$i \leftarrow i+2$

**FinTQ**

ecrire ('la phrase après permutation est : ', p)

#### **Fin**

Programme

```
File Edit Search Run Compile Debug Tools  
[ ] EX01_3.PAS  
program exo9;  
var p:string;  
    c:char ;  
    i:integer;  
begin  
writeln('entrez une phrase');  
readln(p);  
i:=1;  
while i < length(p) do  
begin  
c:=p[i];  
p[i]:=p[i+1];  
p[i+1]:=c;  
i:=i+2;  
end;  
  
writeln('la phrase après permutation est : ',p);  
readln;  
end.
```

9:10

```
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Mak
```

## Exercice 10

Ecrire un programme qui lit un vecteur (tableau) de nombres entiers puis affiche ce vecteur. Calculer la somme des éléments de ce tableau. Trouver le plus grand nombre de ce tableau

### Algorithme

**Algorithme** exo10

**Var** tab : tableau [1,7] d'entier

i ,s, max : entier

**Debut**

**Pour** i de 1 à10 **faire**

Lire (tab[i])

**Finpour**

**Pour** i de 1 à10 **faire**

ecrire (tab[i])

**Finpour**

s ← 0

**Pour** i de 1 à10 **faire**

s ← s+ tab[i]

**Finpour**

Ecrire ('la somme des éléments du tableau est : ', s)

Max=tab[1]

**Pour** i de 2 à10 **faire**

**Si** tab[i] > max **Alors**

Max ← tab[i]

**finsi**

**Finpour**

ecrire ('le maximum est : ', max )

**Fin**

## Programme

```
File Edit Search Run Compile Debug Tools Options
[ ] EXD10.PAS
program exo10;
var tab:array [1..7] of integer;
    i, s, max : integer;
begin
for i:=1 to 7 do
    readln(tab[i]);
for i:=1 to 7 do
    writeln(tab[i]);
s:=0;
for i:=1 to 7 do
    s:=s+tab[i];
writeln('la somme des éléments du tableau est : ',s);
max:=tab[1];
for i:=2 to 7 do
    if tab[i] > max then
        max:=tab[i];
writeln('le maximum est : ',max);
readln;
end.
* 12:34
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F
```

## Exercice 11

Ecrire un programme qui fait la permutation de deux tableaux (élément par élément) remplis automatiquement par des valeurs comprises entre 5 et 50 (utiliser la fonction *Randomize*)

### Algorithme

**Algorithme** exo10

**Var** tab1, tab2 : tableau [1,7] d'entier

i ,c : entier

**Debut**

radomize

**Pour** i de 1 à10 **faire**

Tab1[i] ← 5+radom(45)

Tab2[i] ← 5+radom(45)

**Finpour**

**Pour** i de 1 à10 **faire**

Ecrire (Tab1[i], ' ')

Ecrire(Tab2[i])

**Finpour**

Ecrire ('la permutation des deux tableaux est :')

**Pour** i de 1 à10 **faire**

c ← tab1[i]

tab1[i] ← tab2[i]

tab2[i] ← c

Ecrire (Tab1[i], ' ')

Ecrire(Tab2[i])

**Finpour**

**Fin**

## Programme

```
File Edit Search Run Compile Debug Tool:
[ ] EX011.PAS
program exo11;
var tab1, tab2 :array [1..7]of integer;
    i, c: integer;
begin
  randomize;
  for i:=1 to 7 do
  begin
    tab1[i]:=5+random(45);
    tab2[i]:=5+random(45);
  end;
  for i:=1 to 7 do
  begin
    write (tab1[i],' ');
    writeln (tab2[i]);
  end;
  writeln('la permutaion des deux tableaux');
  for i:=1 to 7 do
  begin
    c:=tab1[i];
    tab1[i]:=tab2[i];
    tab2[i]:=c;

    write (tab1[i],' ');
    writeln (tab2[i]);
  end;
  readln;
end.
```

1:1

F1 Help F2 Save F3 Open Alt+F9 Compile F9

## Exercice 12

Refaire l'exercice 05 (série de FIBONACCI) en utilisant un tableau de 20 éléments.

### Algorithme

**Algorithme** exo5

**Var** i : entier

Tab : tableau [1..20] d'entier

**Debut**

Tab[1] ← 1

Tab[2] ← 1

Ecrire('u1=', Tab[1])

Ecrire('u2=', Tab[2])

**Pour** i de 3 à n **Faire**

Tab[i] ← Tab[i-1] + Tab[i-2]

ecrire ('u', i, ' = ', tab[i] )

**Finpour**

**Fin**

### Programme

```
File Edit Search Run Compile Deb
[ ] EX010
program exo12;
var tab:array [1..20] of integer;
    i : integer;
begin
  tab[1] :=1;
  tab[2] :=1;
  writeln('u1=',tab[1]);
  writeln('u2=',tab[2]);
  for i:=3 to 20 do
    begin
      tab[i]:=tab[i-1]+tab[i-2];
      writeln('u',i,' = ',tab[i]);
    end;
  readln;
end.
```

1:14

F1 Help F2 Save F3 Open Alt+F9 Comp

## Exercice 13

Ecrire un programme qui fait le tri d'un vecteur dans un ordre croissant puis dans l'ordre décroissant.

### Algorithme

<p><b>Algorithme</b> exo13</p> <p><b>Var</b> i,c,j,indmin,nim : entier Tab : tableau [1.7] d'entier</p> <p><b>Debut</b></p> <p><b>Pour</b> i de 1 à 7 <b>faire</b> Lire (tab[i])</p> <p><b>Finpour</b></p> <p><b>Pour</b> i de 1 à 7 <b>faire</b> Min=tab[i] Indmin ← i</p> <p><b>Pour</b> j de i+1 à 7 <b>faire</b> <b>Si</b> tab[j] &lt; min <b>Alors</b> Min ← tab[j] Indmin ← j <b>finsi</b></p> <p><b>Finpour</b> c ← Tab[i] Tab[i] ← min Tab[indmin] ← c</p> <p><b>Finpour</b></p> <p><b>Pour</b> i de 1 à 7 <b>faire</b> Lire (tab[i])</p> <p><b>Finpour</b></p> <p><b>Fin</b></p>	<p><b>Algorithme</b> exo13</p> <p><b>Var</b> i,c,j, indmax, max : entier Tab : tableau [1.7] d'entier</p> <p><b>Debut</b></p> <p><b>Pour</b> i de 1 à 7 <b>faire</b> Lire (tab[i])</p> <p><b>Finpour</b></p> <p><b>Pour</b> i de 1 à 7 <b>faire</b> Max=tab[i] Indmax ← i</p> <p><b>Pour</b> j de i+1 à 7 <b>faire</b> <b>Si</b> tab[j] &gt; max <b>Alors</b> Max ← tab[j] Indmax ← j <b>finsi</b></p> <p><b>Finpour</b> c ← Tab[i] Tab[i] ← max Tab[indmax] ← c</p> <p><b>Finpour</b></p> <p><b>Pour</b> i de 1 à 7 <b>faire</b> Lire (tab[i])</p> <p><b>Finpour</b></p> <p><b>Fin</b></p>
---	---

# Programme

```
File Edit Search Run Compile
[ ]----- NO
program exo;
var tab : array[1..7] of integer;
    i, j, min, indmin, c : integer;
begin
for i:=1 to 7 do
  readln(tab[i]);
for i:=1 to 7 do
begin
min:=tab[i]; indmin:=i;
  for j:=i+1 to 7 do
  begin
    if min > tab[j] then
    begin
      min:=tab[j] ;
      indmin:= j;
    end;
  end;
  c:= tab[i];
  tab[i]:= min;
  tab[indmin]:=c;
end;
end;
```

```
for i:=1 to 7 do
  writeln(tab[i]);
readln;
end.
*----- 9:24 -----
F1 Help F2 Save F3 Open Alt+F9 C
```

```
File Edit Search Run Compile
[ ]----- NO
program exo;
var tab : array[1..7] of integer;
    i, j, max, indmax, c : integer;
begin
for i:=1 to 7 do
  readln(tab[i]);
for i:=1 to 7 do
begin
max:=tab[i]; indmax:=i;
  for j:=i+1 to 7 do
  begin
    if max < tab[j] then
    begin
      max:=tab[j] ;
      indmax:= j;
    end;
  end;
  c:= tab[i];
  tab[i]:= max;
  tab[indmax]:=c;
end;
end;
```

```
for i:=1 to 7 do
  writeln(tab[i]);
readln;
end.
*----- 21:17 -----
F1 Help F2 Save F3 Open Alt+F9 C
```

## Exercice 14

Ecrire un programme qui lit une matrice (4 x 4) de nombres entiers puis affiche cette matrice.

### Algorithme

**Algorithme** exo10

**Var** mat : matrice [1..4, 1..4]

d'entier

i ,s, max : entier

**Debut**

**Pour** i de 1 à4 **faire**

**Pour** j de 1 à 4 **faire**

Lire (mat[i,j])

**Finpour**

**Finpour**

**Pour** i de 1 à4 **faire**

**Pour** j de 1 à 4 **faire**

ecrire (mat[i,j])

**Finpour**

**Finpour**

**Fin**

### Programme

```
File Edit Search Run Compile Debu
[ ] NONAME
program exo14;
uses crt;
var mat : array[1..4,1..4] of integer;
    i, j, max, indmax, c :integer;
begin
for i:=1 to 4 do
begin
for j:=1 to 4 do
readln(mat[i,j]);
end;
{clrscr;}
for i:=1 to 4 do
begin
for j:=1 to 4 do
begin
write(mat[i,j], ' ');
end;
writeln
end;
readln;
end.
* 1:14
F1 Help F2 Save F3 Open Alt+F9 Comp
```