

**NF01**

**Travaux dirigés**



# TD N°1 : numération, diagrammes de Conway

## 1. Changements de base

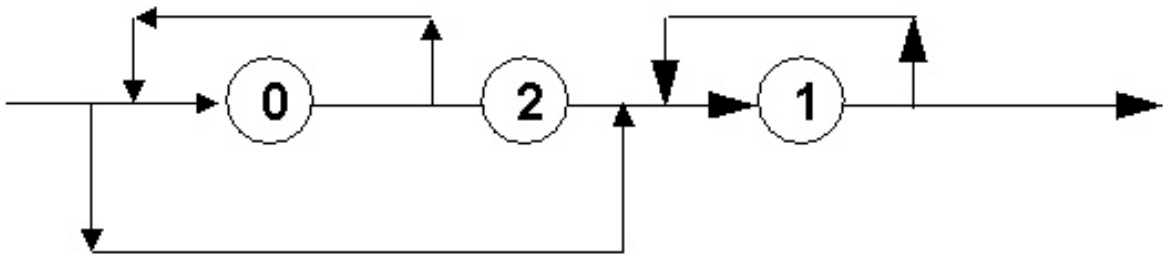
Convertir :

- en décimal les nombres  $101101_2, 563767_8, 2A56E_{16}$ ,
- en binaire les nombres  $173_{10}, 17657_8, 1AF3C_{16}$ ,
- en hexadécimal les nombres  $364_{10}, 11011100_2, 76472_8$ ,
- en octal les nombres  $3567_{10}, 3AE5_{16}, 1010111_2$ .

## 2. Diagrammes de Conway

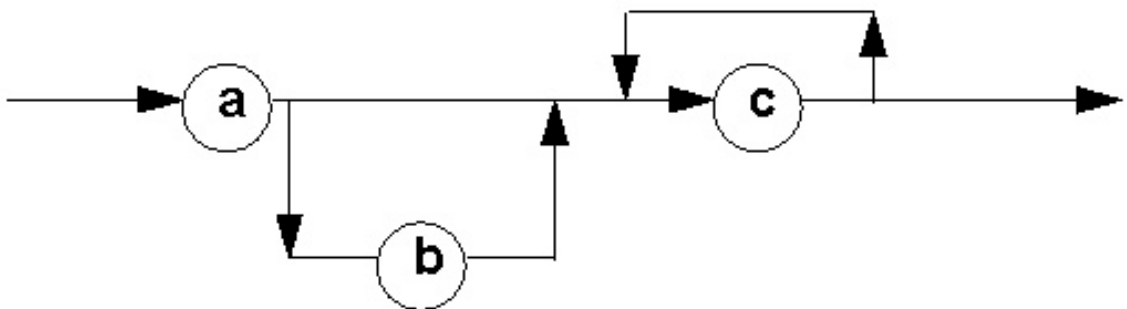
2.1 Soit le langage constitué des mots suivants :  $b, ab, aab, aaab, a...ab$ . Donner une formule pour définir ce langage. Ecrire le diagramme de Conway correspondant.

2.2 Soit le diagramme suivant :



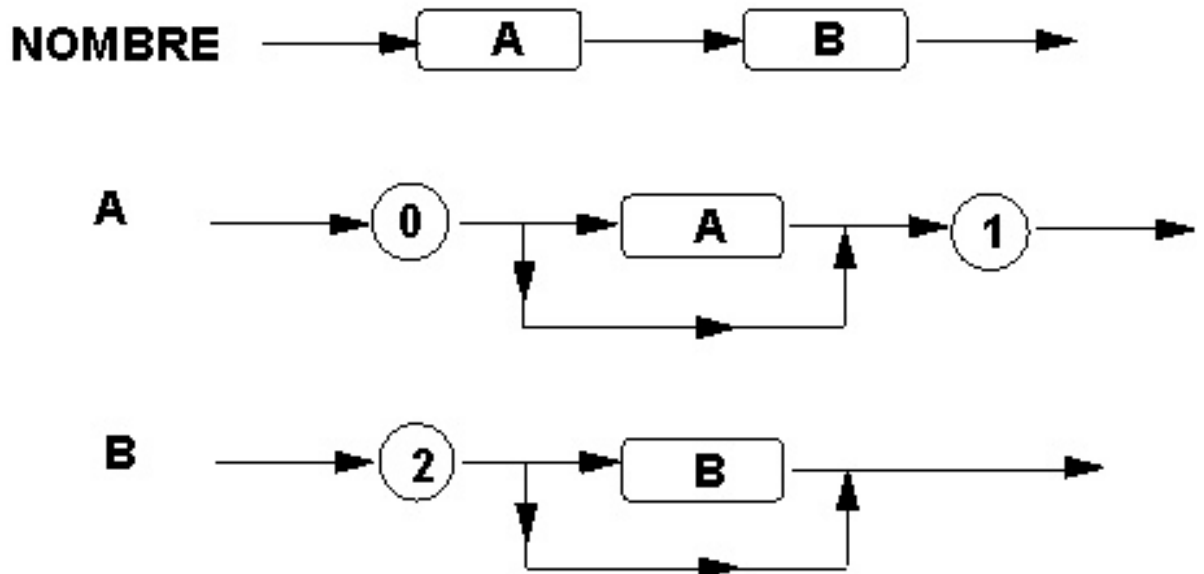
Définir le langage correspondant à ce diagramme par une formule. Donner des exemples de phrases acceptées et de phrases refusées par ce langage.

2.3 Définir le langage exprimé par le diagramme suivant :



Donner des exemples de phrases acceptées par ce langage.

2.4 Définir le langage exprimé par le diagramme suivant :



Donner des exemples de phrases acceptées par ce langage ?

2.5 Dessiner les diagrammes de CONWAY qui permettent de vérifier la grammaire suivante :  
 $aa+, ba+, ab+, bb+, aa-, ab-, ba-, bb-, aaa++, aba+-, bab--$

Il s'agit donc d'une suite de signes  $a$  ou  $b$ , suivie d'une suite de signes  $+$  ou  $-$ .  
 On remarquera cependant qu'il y a toujours un signe de type  $a$  ou  $b$  de plus que de signe  $+$  ou  $-$ .

## TD N°2 : algorithmes

1. Écrire un algorithme qui affiche le minimum de deux nombres entrés au clavier. Étendre cet algorithme au cas de trois nombres.
  
2. Écrire un algorithme qui calcule la surface et le volume d'une sphère à partir du rayon qui sera lu au clavier.
  
3. Écrire un algorithme qui calcule la résistance équivalente à 3 résistances montées en parallèle ou en série. Les données fournies en entrée seront le type de montage et la valeur des trois résistances. Le programme devra afficher de manière claire la valeur de la résistante équivalente.
  
4. Écrire une séquence d'instructions permettant de permuter la valeur de deux variables x et y.
  
5. Écrire un algorithme qui affiche la valeur absolue d'un nombre entré au clavier.
  
6. Déterminer un algorithme qui calcule l'heure d'arrivée d'un train à partir de l'heure de départ et de la durée du trajet. Chaque donnée d'entrée sera fournie sous forme de trois nombres précisant les heures, minutes et secondes. Le résultat sera aussi indiqué sous cette même forme.
  
7. Écrire un algorithme qui lit un nombre entré au clavier par l'utilisateur et qui détermine si ce nombre est ou non compris entre 10 (inclus) et 20 (inclus).



## TD N°3 : expressions, structures conditionnelles

1. Montrer en utilisant la définition syntaxique d'une expression, comment se décomposent les expressions suivantes :

$(a+b)/c$

$a*\sin(x-y)$

$a+b-c-d/e$

$a = b \text{ and } c$

2. Calcul de remise :

Ecrire un programme qui détermine un montant net à partir d'un montant brut (entré par l'utilisateur) en appliquant une remise de :

- 5% si le montant brut est compris entre 200 et 500 €
- 10% si le montant brut est supérieur à 500 €

3. On trouve dans un programme Pascal l'instruction suivante dans laquelle les variables a, b et c sont booléennes et la variable x est entière :

```
if a then if not (b) then
    if not (a) then x :=0 else if c then x :=1
else
    x :=0
else if not (b) then x :=1
else if not (c) then x :=0 else x :=1
    else if c then
x :=1 else x :=0 ;
```

Réécrire cette instruction en respectant la présentation conseillée en cours.  
Essayer de simplifier cette instruction.

4. Ecrire un algorithme qui permet de déterminer si une année est bissextile. On rappelle que :

- Les années bissextiles sont divisibles par 4
- Toutefois, les années divisibles par 100 ne sont pas bissextiles
- A l'exception des années divisibles par 400 qui, elles, le sont.

On cherchera à résumer ces conditions en une seule expression booléenne.

5. Tarif d'entrée dans un musée (Extrait du médian P12)

Ecrire un programme permettant au guichetier d'un musée de calculer le tarif d'entrée dans le musée. Le programme demandera au guichetier le prix d'entrée de base (selon l'exposition visitée), l'âge du visiteur et si ce visiteur possède une carte d'étudiant ou de chômeur. Ensuite, selon ces données, le programme pourra calculer une réduction éventuelle. Les réductions possibles seront calculées de la façon suivante :

- si le visiteur a moins de 18 ans, l'entrée est gratuite ;
- si le visiteur a entre 18 et 25 ans, et s'il est étudiant, il a une réduction de 50%, mais s'il n'est pas étudiant la réduction n'est que de 30% ;
- si le visiteur a plus de 25 ans, mais est quand même étudiant, alors il a une réduction de 30% ;
- si le visiteur a plus de 60 ans, il paie demi-tarif ;
- si le visiteur est chômeur, la visite est gratuite.

Attention, il n'est pas possible de cumuler les réductions.





## TD N°4 : structures itératives

1. Écrire un algorithme qui lit une série de nombres au clavier et qui s'arrête à la première valeur négative.
2. Écrire un algorithme qui calcule le produit d'une série de nombres lus au clavier. On supposera que l'on entrera la valeur 0 pour indiquer la fin de la série, cette valeur n'étant pas prise en compte.
3. Écrire un algorithme qui lit 10 nombres et indique la valeur du plus petit.
4. Écrire un algorithme qui lit 10 nombres et indique le nombre de valeurs négatives
5. Écrire un algorithme, puis un programme, qui affiche les 100 premiers termes de la suite  $U_n$  définie par :  $U_0 = 2$  et  $U_{n+1} = U_n + 7$ .
6. Écrire un algorithme qui affiche à l'écran la pyramide suivante (nombre de lignes variable) :

```
      1
     1 2 1
    1 2 3 2 1
      .
      .
      .
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
```

N.B. : on cherchera à écrire également les programmes Pascal correspondant.



## TD N°5 : tableaux

1. Ecrire un algorithme, puis un programme Pascal, qui remplit les 100 éléments d'un tableau avec la valeur du carré de chaque indice. Par exemple, le premier élément aura comme valeur 1, le deuxième 4, et le dixième 100... Le programme devra ensuite afficher ce tableau à l'écran.
2. Ecrire un algorithme puis un programme qui lit 12 nombres entrés au clavier, en fait la moyenne et affiche les nombres supérieurs à cette moyenne.
3. On peut représenter un vecteur de l'espace vectoriel  $R_n$  à l'aide d'un tableau de  $n$  réels.
  - a) Ecrire un algorithme qui lit deux vecteurs de  $R_{10}$ , calcule leur produit scalaire et affiche les deux vecteurs et leur produit scalaire.
  - b) Écrire un algorithme qui lit deux vecteurs de  $R_7$ , calcule leur somme et affiche cette somme.
4. La moyenne olympique d'un ensemble de nombres est obtenue en faisant la moyenne des nombres qui restent une fois supprimés le plus grand et le plus petit.  
Ecrire un algorithme, puis un programme, qui lit une série de nombres (on supposera que tous les nombres sont différents) et affiche leur moyenne olympique.
5. On considère des matrices carrées de nombres entiers de type :  
MatriceCarree = array[1 .. NMAX, 1 .. NMAX] of integer ; { NMAX étant une constante }  
Ecrire un programme permettant de :
  - a) Calculer la moyenne des valeurs d'une matrice de ce type
  - b) Vérifier si une matrice carrée est magique (la somme des valeurs sur chaque ligne, sur chaque colonne et sur les deux diagonales est la même).

*Exemple de matrice magique :*

14	5	17
15	12	9
7	19	10

*Remarque :* dans un premier temps, on n'écrira pas la partie du programme correspondant à la saisie des valeurs de la matrice par l'utilisateur.



## TD N°6 : structures itératives (Annales de médians)

**Examen n°1 :** Soit le programme Pascal suivant (qui est très mal présenté !) :

```

Program inconnu;
var i,j : integer; a,d,c,b,m : longint;
begin read(i,j);m:=0; a:=1;
for b:=1 to j do a:=a*i; d:=1;
writeln('étape 1 : valeur de a et de d : ',a:6,d:6);
for c:=1 to a do
d:=d*c; writeln('étape 2 : valeur de a et de d : ', a:6,d:6);
d:=1;a:=0;for b:=1 to j do d:=d*b;
writeln('étape 3 : valeur de a et de d : ', a:6,d:6);
a:=1; for b:=1 to d
do a:=a*i; writeln('étape 4 : valeur de a et de d : ', a:6,d:6);
end.

```

Q1 : Remplir les tableaux ci-dessous, afin d'indiquer les valeurs des variables à différentes étapes du programme, avec les données proposées en tête de tableau.

Q2 : Réécrire ce programme en donnant des noms d'identificateurs signifiant quelque chose à la lecture du programme, en le présentant correctement et en y ajoutant des commentaires.

Q3 : Identifier les parties communes du programme et essayer de décrire ce qu'elles font.

Q4 : A quoi correspond chaque `writeln` lors des 4 étapes ? Quel est le calcul présenté ?

Q5 : Que fait ce programme dans son ensemble ? A quoi correspond-il ? Expliquer.

i=2 et j=3	a	b	c	d
<b>étape 1</b>				
<b>étape 2</b>				
<b>étape 3</b>				
<b>étape 4</b>				

i=3 et j=2	a	b	c	d
<b>étape 1</b>				
<b>étape 2</b>				
<b>étape 3</b>				
<b>étape 4</b>				

**Examen n°2 :** Ecrire un **algorithme** qui demande une valeur V (réelle) comprise entre 1 et 2, et qui calcule le plus petit N (entier) tel que:  $1 + 1/2 + 1/3 + \dots + 1/N > V$

**Examen n°3:** Que fait le programme suivant ? 10 mn

```

program abc;
var a, b, c, temp : integer;
begin write('a='); readln(a);
write('b='); readln(b); write('c='); readln(c);
if b > a then begin temp := a;
a := b; b := temp;end;
if c > a then begin
temp := a; a := c; c := temp;end;
if c > b then begin
temp := b; b := c;
c := temp;end;
writeln(a, b, c); end.

```

Donner un exemple de données d'entrée et de sortie.

**Examen n°4**

Dans chacun des cas suivants, écrire un programme qui affiche année par année l'évolution d'un capital :

- a) pour les 10 prochaines années
- b) jusqu'à ce que le capital de départ ait doublé

Justifier à chaque fois votre choix de boucle.

*Remarques :*

- L'utilisateur devra entrer le capital initial  $C_0$  et le taux de placement annuel  $t$  ( $0 < t < 1$ ).
- Si à l'année  $n$  le capital vaut  $C$ , à l'année  $n+1$  le capital vaudra :  
$$C * (1+t).$$

**Examen n°5**

On souhaite simuler le fonctionnement d'une borne interactive à l'entrée d'un cinéma. Cette borne doit permettre à un client de connaître le tarif exact du film qu'il veut voir.

Les règles sont les suivantes :

- Le tarif normal est de 5 €.
  - Dans les salles 1 et 2, le tarif est majoré de 10 % pour tous les clients.
  - Dans la salle 3, si le client est :
    - un étudiant, il bénéficie d'une réduction de 10 %
    - un enfant de moins de 12 ans, il bénéficie d'une réduction de 50%
    - un militaire, il bénéficie d'une réduction de 5%
- Pour tous les autres clients, la place est au tarif normal.

- 1) Indiquer les données que l'utilisateur devra entrer.
- 2) Écrire le programme permettant de déterminer le tarif en fonction de ces données.

## TD N°7 : procédures et fonctions

1. Écrire une procédure à trois paramètres entiers qui fait la somme des deux premiers et range cette valeur dans le troisième.
2. Transformer la procédure précédente en fonction. Le troisième paramètre devient le retour de la fonction.
3. Écrire une procédure *ordre* qui range par ordre croissant les valeurs de ses trois paramètres (après l'appel `ordre(a,b,c)`, les valeurs des 3 variables doivent vérifier  $a \leq b \leq c$ ).
4. Quelle sera la sortie du programme suivant:

```
program p1;
var a,b,c : integer;

procedure spl (x,y:integer);
var z : integer;
begin
  z := x;
  x := y;
  y := z;
end;

begin
  a := 10;
  b := 20;
  c := 30;
  spl (a,b);
  writeln (a,b,c)
end.
```

5. Donner et justifier brièvement la sortie du programme suivant :

```
program p2;
var
  x : integer;

  procedure a(y :integer) ;
  begin
    x := x+1 ; y := y+2 ;
  end ;

  procedure b(var y : integer) ;
  var x : integer ;
  begin
    x := x+3 ; y := y+4 ;
  end ;

begin
  x := 0 ; writeln(x) ;
  a(x) ; writeln(x) ;
  b(x) ; writeln(x) ;
end.
```

6. Donner et justifier brièvement la sortie du programme suivant. Commenter le résultat. Quel nom porte l'effet constaté ?

```
program p2 ;
var i, count : integer ;

    procedure incrementer (var x : integer) ;
    begin
        x := x+1 ; count := count+1 ;
    end ;

begin
    i :=10 ; count :=0 ;
    incrementer(i) ; writeln(i,count) ;
    incrementer(count) ; writeln(count) ;
end.
```



## TD N°8 : procédures et fonctions (suite)

1. Écrire une fonction qui renvoie la position de la première occurrence de la valeur 12 dans un tableau de 10 entiers entré en paramètre. Le programme affichera 0 comme position si ce nombre 12 n'existe pas. Ecrire un même exemple d'utilisation. Même question si on recherche la dernière occurrence de 12.
  
2. Recherche du maximum :
  - 2.1. Ecrire une fonction qui renvoie la valeur maximum d'un tableau d'entiers. La dimension maximum de ce tableau sera 1000, mais la dimension réelle ce tableau, inférieure ou égale à 1000, sera transmise en paramètre. On donnera un exemple d'utilisation de cette fonction.
  - 2.2. Même question, mais cette fois la fonction devra renvoyer l'indice de la valeur maximale.
  
3. Écrire une procédure permettant de lire des caractères entrés au clavier, et de les transformer en des nombres entiers (1 pour 'A', 2 pour 'B'...). La saisie se termine lorsque l'utilisateur frappe la lettre '\$'.
  
4. Écrire une fonction pgcd qui donne le PGCD de deux nombres x et y.
  
5. Gestion d'un stock de DVD (extrait de l'examen final 2008)

Pour écrire un programme permettant de gérer le stock d'un magasin de vente de DVD, on définit deux tableaux, tabDVDFrançais et tabDVDAnglais, le premier correspondant aux DVD en français et le second aux DVD en anglais. Chaque DVD est représenté par un enregistrement comprenant le nom du film, le genre (fiction, policier, documentaire, action, jeunesse, horreur, thriller, drame, comédie), l'année de sortie, le nombre d'exemplaires, le prix de vente.

Ecrire en Pascal :

- 1) une procédure de saisie du stock de DVD.  
Un paramètre (formel) de cette procédure sera tabDVD. Il pourra recevoir un des deux tableaux d'enregistrements : tabDVDFrançais ou tabDVDAnglais.
  
- 2) une procédure ou fonction permettant d'effectuer des remises de la façon suivante :
  - Un paramètre (formel) de cette procédure sera tabDVD. Il pourra recevoir un des deux tableaux d'enregistrements : tabDVDFrançais ou tabDVDAnglais.
  - La remise sera de 20% sur les DVD de genre « fiction » dont l'année de sortie est inférieure strictement à 2000. Cette remise sera faite en modifiant directement le prix des DVD dans le tableau d'enregistrements.
  - Le nombre de DVD ayant fait l'objet d'une remise sera comptabilisé. Ce nombre tiendra compte du nombre d'exemplaire pour chaque DVD.



## TD N°9 : enregistrements

1. Définir le type date permettant de représenter une date comportant le numéro du jour et le numéro du mois (14 7, 25 12, 31 12, ...).

Ecrire la procédure `jour_du_lendemain` à deux paramètres de type date qui détermine la date du lendemain d'un jour donné. On supposera que l'année n'est pas bissextile.

Ecrire un exemple d'utilisation de cette procédure.

2. On dispose d'un tableau d'enregistrements ayant la structure suivante : nom, prénom, âge.

Ecrire les procédures et fonctions suivantes :

- 2.1. Stockage de l'ensemble des étudiants dans le tableau
- 2.2. Recherche d'un étudiant
- 2.3. Ajout d'un étudiant
- 2.4. Suppression d'un étudiant
- 2.5. Modification d'un étudiant



## TD N°10 : fichiers

1. Ecrire un programme qui écrit dans un fichier, dont chaque article est un entier, une série de nombres tapés au clavier.
2. Ecrire un programme qui imprime la moyenne des nombres contenus dans le fichier précédent.
3. On dispose d'un fichier dont chaque article a la structure suivante :

```
type eleve = record
    nom : string (20) ;
    prenom : string (20) ;
    math : real ;
    physique : real ;
    français : real
end ;
```

Ecrire un programme qui exploite ce fichier pour afficher le tableau suivant

Nom	Math	Physique	Français	Moyenne
Dupont	12.50	10.00	8.50	10.33
Durand	6.50	9.50	8.00	8.00
...				
Moyenne	9.32	11.39	10.48	10.40

On supposera que le nombre d'élèves est inférieur ou égal à 100.

4. Un supermarché désire créer un fichier contenant la liste des prix de tous les articles disponibles en rayon.

Ecrire un programme qui écrit dans un fichier la liste des articles avec leur prix (entrés au clavier). On désire ensuite relire ce fichier et calculer le prix moyen des articles en rayon. On aimerait, pour conclure, rechercher le nombre d'articles pour lesquels le prix est supérieur à un tarif entré par l'utilisateur. Par exemple, on voudrait savoir combien d'articles sont vendus à plus de 1 000 euros.



## TD N° 11 : fichiers texte

1. Ecrire un algorithme permettant de transformer un fichier de texte en remplaçant toutes les lettres minuscules par des majuscules et en supprimant tous les accents. Le résultat doit être un nouveau fichier de texte.
  
2. Ecrire un programme qui lit un texte contenu dans un fichier texte et retourne la fréquence relative de chacune des 26 lettres de l'alphabet. La fréquence relative d'une lettre est le nombre d'occurrence de cette lettre, qu'elle soit minuscule ou majuscule, divisée par le nombre de lettres contenues dans le texte.
  
3. Ecrire un programme qui lit un texte contenu dans un fichier texte et retourne le nombre de mot contenus dans ce texte. On considère que les séparateurs de mots peuvent être les caractères "espaces", "virgule", "point" et "passage à la ligne" et que plusieurs séparateurs peuvent se suivre.





## TD N° 12 : récursivité

1. Trouver une définition récursive du PGCD de deux entiers et écrire l'algorithme puis la procédure correspondante.
2. Trouver un algorithme récursif permettant d'inverser une chaîne de caractères ('NF01' devient '10FN', 'bonjour' devient 'ruojnob'...).
3. En déduire une fonction booléenne permettant d'indiquer si un mot est un palindrome (identique à l'endroit et à l'envers, par exemple : 'noyon' ou 'rever').
4. Ecrire une version récursive de l'algorithme de **recherche dichotomique**. On rappelle que la recherche dichotomique est une recherche par approximations successives dans une liste ordonnée (triée par ordre croissant de ses valeurs). On compare l'élément recherché à celui situé en milieu de liste. S'il est plus grand, on recommence avec la moitié supérieure de la liste, sinon avec la moitié inférieure de la liste, jusqu'à convergence vers l'élément recherché (égalité = succès), ou jusqu'à ce qu'il n'y ait plus aucun élément à comparer (liste vide). Dans ce cas, l'élément recherché était absent de la liste. Ecrire un programme dichotomie permettant d'effectuer une recherche dichotomique dans un tableau de 100 entiers.



## TD N° 13 : tri, fusion de fichiers

1. **Tri par sélection.** On dispose de  $n$  valeurs numériques rangées dans un tableau. Il s'agit de ranger par ordre croissant ces  $n$  valeurs, soit dans le même tableau, soit dans un autre tableau. Par exemple, si on dispose au départ des 6 valeurs 7, 3, 1, 2, 5, 4, ils devront être rangés à la fin dans l'ordre 1, 2, 3, 4, 5, 7. Voici un algorithme simple, appelé **tri par sélection**, réalisant ce tri :

On cherche le plus petit nombre et on le permute avec le nombre placé en première position, on cherche le plus petit des nombres restants et on le permute avec le nombre placé en seconde position,... Avec notre exemple cela donnerait :

7	3	1	2	5	4
1	3	7	2	5	4
1	2	7	3	5	4
1	2	3	7	5	4
1	2	3	4	5	7
1	2	3	4	5	7

L'algorithme demandé aura la structure suivante : lecture d'un tableau d'entiers, tri de ce tableau et affichage des nombres ainsi triés.

### 2. Fusion de deux fichiers séquentiels

- 2.1. Définir le type personnel comme une structure d'enregistrement contenant le nom, l'âge, le sexe et la catégorie socio-professionnelle qui peut-être ouvrier, technicien, cadre, commerçant et profession libérale.
- 2.2. Ecrire un programme qui à partir de deux fichiers séquentiels constitués d'enregistrements de type personnel qui sont déjà ordonnés suivant les noms crée un troisième fichier de même type contenant l'ensemble ordonné des employés présents dans les deux premiers fichiers. (on supposera que chaque fichier se termine par un enregistrement supplémentaire caractérisé par un nom de longueur nulle).