

10) Tableaux

Ecrire une fonction *Affichage* prenant en argument un tableau T de N réels et qui effectue l'affichage de ce tableau.

Ecrire une fonction *Lecture* qui prend un tableau T de N réels en argument et qui remplit le tableau T avec N réels lus au clavier.

Ecrire le programme principal permettant d'utiliser ces 2 fonctions pour un tableau de 50 réels au maximum.

Ecrire une fonction *Moyenne* prenant en argument un tableau T de N réels et qui calcule la moyenne des éléments de ce tableau.

11) Tri de tableaux

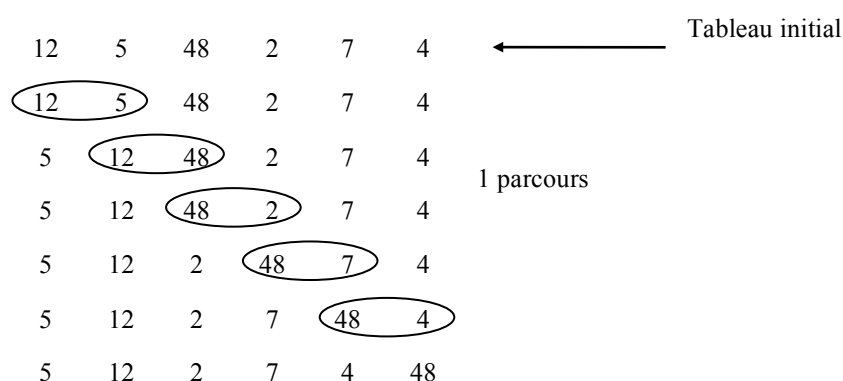
La fonction suivante permet de mettre dans un tableau n nombres aléatoires.

```
void alea(int a[],int n)
{
  int i;
  srand(1);
  for (i=0;i<n;i++) a[i]=rand()/100;
}
```

Écrire une fonction permettant d'afficher les n éléments du tableau.

Tri à bulles

On effectue un certain nombre de parcours du tableau à classer, un parcours consiste à aller d'un bout à l'autre du tableau en effectuant la comparaison de deux éléments successifs et en les permutant s'ils ne sont pas classés. Cette comparaison remonte dans le tableau comme une bulle, en entraînant l'extremum.

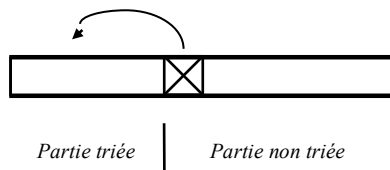


A l'issue du premier parcours, l'élément 48 est à sa place.

On peut améliorer le tri par bulles en limitant les parcours à la partie non triée du tableau : après un parcours, l'élément maximal est rangé, le deuxième parcours peut donc être limité à $n-1$ positions. Après deux parcours, les deux plus grands éléments sont classés, le troisième parcours peut être limité à $n-2$ positions etc...

Écrire la procédure *tri_bulles(n,A)* permettant de trier par la méthode bulles le tableau A .

Tri par insertion



Tant que la partie non triée n'est pas vide, on prend l'élément frontière avec la partie triée et on l'insère à sa place dans cette dernière.

L'insertion de l'élément frontière est effectuée par décalages ou permutations successives à l'aide de la procédure $insérer(i,A)$ qui permet d'insérer le i ème élément du tableau A à sa place.

Écrire la procédure $tri_insertion(n,A)$ permettant de trier par insertion le tableau A .

12) Tableau de contingence

Étant donné le tableau de contingence ci-dessous :

	1	...	j	...	q	
1	n_{11}	...	n_{1j}	...	n_{1q}	$n_{1.}$
i	n_{i1}	...	n_{ij}	...	n_{iq}	$n_{i.}$
p	n_{p1}	...	n_{pj}	...	n_{pq}	$n_{p.}$
	$n_{.1}$...	$n_{.j}$...	$n_{.q}$	n

avec

$$n_{i.} = \sum_{j=1}^q n_{ij}$$

$$n_{.j} = \sum_{i=1}^p n_{ij}$$

$$n = \sum_{i=1}^p n_{i.} = \sum_{j=1}^q n_{.j}$$

$$\text{Calculer : } p = \frac{\prod_i n_{i.}! \prod_j n_{.j}!}{n! \prod_i \prod_j n_{ij}!}$$

Exemple : pour le tableau

3	6
4	7

$$\text{ on a } \frac{(9!11!)(7!13!)}{20!3!4!6!7!} = 0,35758$$

Écrire un programme structuré en fonctions permettant de calculer p .