
Travaux dirigés 6 : lecture de données au clavier.

1 Lecture de données utilisateur entrées à partir du clavier

La fonction `scanf`, déclarée dans la bibliothèque `stdio` permet d'affecter à des variables des valeurs saisies au clavier par l'utilisateur du programme. Les types de données à lire sont précisés avec les mêmes chaînes de format que pour `printf`, à quelques exceptions près (par exemple `"%g"` pour afficher un double et `"%lg"` pour lire un double).

1. Que fait le programme suivant ?

```
1  /* declaration de fonctionnalites supplementaires */
2  #include <stdlib.h> /* EXIT_SUCCESS */
3  #include <stdio.h> /* printf, scanf */
4
5  /* declarations des constantes et types utilisateurs */
6
7  /* declarations des fonctions utilisateurs */
8
9  /* fonction principale */
10 int main()
11 {
12     int a;
13     double b;
14     char c;
15
16     printf("Entrez un nombre entier puis un nombre réel puis un caractère : ");
17
18     scanf("%d",&a);
19     scanf("%lg",&b);
20     scanf(" %c",&c);
21
22     printf("Vous avez saisi %d puis %g puis %c.\n",a,b,c);
23
24     return EXIT_SUCCESS;
25 }
26
27 /* definitions des fonctions utilisateurs */
```

2. Faire la trace du programme en considérant que l'utilisateur saisit au clavier : 1 puis "entrée", 12.2 puis "entrée" et 'c' puis "entrée" .

1.1 Calcul de la moyenne d'une série d'entiers saisie par l'utilisateur

Écrire un programme qui demande à l'utilisateur combien d'entiers composent sa série, lit la série d'entiers et affiche la moyenne de valeurs de la série. L'ensemble de la série ne doit pas être stockée en mémoire.

1.2 Initialisation d'un tableau par l'utilisateur

Écrire un programme qui déclare un tableau d'entiers de taille arbitraire TAILLE (une constante symbolique) et qui demande à l'utilisateur de saisir au clavier les valeurs des cases du tableau. Afficher le tableau.

2 Les sujets du contrôle de TP

Sujet 1. Soit un tableau d'entiers relatifs initialisé à des valeurs de votre choix et dont la taille sera fixée par une constante symbolique N. Écrire un programme qui :

1. affiche le contenu du tableau ;
2. affiche la valeur absolue des éléments du tableau ;
3. calcule et affiche la somme des valeurs absolues des éléments du tableau.

Le programme doit fonctionner correctement pour n'importe quelle initialisation du tableau.

Sujet 2. Soit une constante symbolique N valant 3. Écrire un programme qui :

1. affiche le carré de côté N suivant :

```
(0,0) (0,1) (0,2)
(1,0) (1,1) (1,2)
(2,0) (2,1) (2,2)
```

2. puis affiche le carré de côté N suivant :

```
***** (0,1) (0,2)
(1,0) ***** (1,2)
(2,0) (2,1) *****
```

3. puis affiche le triangle de côté N suivant :

```
(0,0) (0,1) (0,2)
      (1,1) (1,2)
        (2,2)
```

Le programme doit fonctionner correctement pour n'importe quelle valeur positive de N strictement inférieure à 10.

Sujet 3. Soit un tableau d'entiers initialisé à des valeurs de votre choix et dont la taille sera fixée par une constante symbolique N. Soit un entier x initialisé à une valeur de votre choix. Écrire un programme qui :

1. affiche les valeurs du tableau et celle de x ;
2. compte le nombre cases du tableau contenant la valeur de x (c'est à dire le nombre d'occurrences de x dans le tableau) et affiche ce nombre à l'écran.

Le programme doit fonctionner correctement pour n'importe quelle initialisation du tableau et n'importe quelle valeur de x .

Sujet 4. Soit un tableau d'entiers relatifs initialisé à des valeurs de votre choix et dont la taille sera fixée par une constante symbolique N. Écrire un programme qui :

1. affiche le contenu du tableau
2. compte le nombre d'entiers positifs ou nuls dans le tableau et affiche ce nombre à l'écran.

Le programme doit fonctionner correctement pour n'importe quelle initialisation du tableau.

Sujet 5. Soit une variable entière n initialisée à une valeur positive. Écrire un programme qui :

1. calcule la factorielle de n (le produit des entiers de 1 à n) et affiche le résultat. Exemple de sortie (pour $n = 3$) :

Factorielle de 3 : $3! = 6$.

2. calcule la somme des factorielles des n premiers entiers positifs. Exemple de sortie (pour $n = 3$) :

Somme des factorielles jusqu'à 3 :
 $1! + 2! + 3! = 9$.

Le programme doit fonctionner pour n'importe quelle valeur positive de n .