
Travaux dirigés 7 : lecture de données au clavier.

1 Lecture de données utilisateur entrées au clavier

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 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 des 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 En TP : le nombre secret

Nous voulons programmer le jeu du nombre à découvrir. Le joueur doit deviner un nombre secret choisit par l'ordinateur entre 0 et `NB_MAX` (une constante du programme). S'il propose un nombre trop grand, l'ordinateur lui répond "Plus petit", s'il propose trop petit, l'ordinateur lui répond "Plus grand". Dans ces deux cas, il est invité à proposer un autre nombre. Le jeu s'arrête quand il devine juste. Un exemple d'exécution de ce jeu pourrait être :

```
Votre choix ?
8
Plus petit.
Votre choix ?
4
Plus petit.
Votre choix ?
2
Vous avez trouvé le nombre secret.
```

1. Proposez un algorithme en français pour le jeu.
2. Traduisez-le en langage C et exécutez-le.
3. Pourquoi préférez-vous une boucle `while` ici ?

Pour rendre le jeu intéressant, l'ordinateur doit choisir le nombre secret *au hasard*. La librairie C standard propose des fonctions renvoyant des nombres pseudo-aléatoires¹ déclarées dans `<stdlib.h>`. L'ordinateur va utiliser la fonction : `int rand()` ; qui renvoie un nombre pseudo-aléatoire entier entre 0 et la constante `RAND_MAX` (égale à 2147483647) inclus. Pour renvoyer un nombre pseudo-aléatoire entre 0 et `NB_MAX`, `NB_MAX` inclus (`NB_MAX << RAND_MAX`), il suffit de calculer le reste de la division entière de `rand()` par (`NB_MAX + 1`), c'est-à-dire le nombre renvoyé par `rand()` modulo (`NB_MAX + 1`) (opérateur `%` en C). `rand` vient de `random` qui veut dire aléatoire en anglais.

Un exemple de programme illustrant l'utilisation de `rand` pour engendrer un nombre pseudo-aléatoire est le suivant :

```
#include <stdlib.h> /* EXIT_SUCCESS, rand, srand */
#include <time.h> /* time */

#define NB_MAX 15 /* nombre secret entre 0 et NB_MAX inclus */

int main()
{
    int nombre_secret; /* nombre secret à deviner */

    /* initialisation du générateur de nombres pseudo-aléatoires */
    srand(time(NULL)); /* à ne faire qu'une fois */

    /* tirage du nombre secret */
    nombre_secret = rand() % (NB_MAX + 1); /* entre 0 et NB_MAX inclus */

    /* manche joueur ... */

    return EXIT_SUCCESS;
}
```

1. http://fr.wikipedia.org/wiki/Générateur_de_nombres_pseudo-aléatoires