
Travaux dirigés 6 : type booléen en C ; structures de contrôle “for”et “while”

1 Évaluation d'expressions booléennes

Soit le programme suivant :

```
#include <stdlib.h> /* EXIT_SUCCESS */  
#include <stdio.h> /* printf */  
  
#define FALSE 0  
#define TRUE 1  
  
/* Declaration de fonctions utilisateurs */  
  
int main()  
{  
    int beau_temps = TRUE;  
    int pas_de_vent = FALSE;  
  
    printf("%d\n",beau_temps && pas_de_vent);  
    printf("%d\n",beau_temps || pas_de_vent);  
    printf("%d\n",! beau_temps || pas_de_vent);  
    printf("%d\n",! (! beau_temps || pas_de_vent) == (beau_temps && ! pas_de_vent));  
  
    return EXIT_SUCCESS;  
}  
  
/* Definition de fonctions utilisateurs */  
  
1. Qu'affiche le programme ?  
2. Modifiez le programme pour qu'il demande la valeur des booléens à l'utilisateur (0 pour  
FALSE, sinon TRUE).
```

2 Boucles *for* ou *while* ?

Ces boucles ont exactement la même sémantique et on peut facilement réécrire l'une en l'autre. Par convention, on préfère utiliser la boucle *for* lorsque l'on connaît le nombre d'itérations à l'avance ; on utilise *while* dans le cas contraire, lorsque le nombre d'itérations n'est pas connu à l'avance. Par exemple, pour faire la somme des éléments d'un tableau on utilisera *for* et pour trouver un élément dans un tableau, ne sachant pas où il se trouve, on choisira *while*. L'intérêt de respecter cette convention est que la lecture d'un programme est facilité en indiquant à quoi sert la boucle.

Résoudre les problèmes suivants en utilisant soit *for*, soit *while*.

2.1 Test d'égalité entre tableaux

Écrire un programme qui teste si deux tableaux de même TAILLE (constante symbolique) ont les mêmes données.

2.2 Nombre d'occurrences dans un tableau

Écrire un programme qui fait initialiser un tableau de taille TAILLE par l'utilisateur, demande à l'utilisateur un entier, et affiche le nombre d'occurrences de l'entier dans le tableau. Des exemples de sorties sont les suivants :

```
Saisissez 4 entiers : -2 -2 -2 3
Compte le nombre d'occurrences de quel entier ?
-2
Il y a 3 occurrences de -2 dans le tableau.
```

```
Saisissez 4 entiers : -2 -2 -2 3
Compte le nombre d'occurrences de quel entier ?
3
Il y a 1 occurrences de 3 dans le tableau.
```

```
Saisissez 4 entiers : -2 -2 -2 3
Compte le nombre d'occurrences de quel entier ?
0
Il y a 0 occurrences de 0 dans le tableau.
```

2.3 Élévation à la puissance

Écrire un programme qui demande à l'utilisateur d'entrer deux nombres entiers x et $n \geq 0$ puis calcule x^n et affiche le résultat.

2.4 Test de primalité

Écrire un programme qui demande à l'utilisateur d'entrer un nombre entier positif n , teste si n est premier puis affiche le résultat.

3 Carré d'étoiles

Écrire un programme qui demande à l'utilisateur d'entrer un nombre entier positif n et affiche un carré creux d'étoiles de côté n . Exemple (l'utilisateur entre 4) :

```
n? 4
*****
* *
* *
*****
```