

Fiche n° 4 de TP

Instructions conditionnelles et alternatives (1)

Objectifs : manipulation des instructions conditionnelles et alternatives ; manipulation des expressions booléennes.

Prérequis : syntaxe des instructions conditionnelles et alternatives ; syntaxe des expressions booléennes.

Travail minimum : exercices 1 à 4.

Exercice 1

Tapez sans y apporter la moindre modification le programme suivant dans le fichier `tp4-ex1.c` :

```
tp4-ex1.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void) {
5     int x;
6     int y;
7     if (scanf("%d%d", &x, &y) != 2) {
8         printf("Erreur : échec lors de la lecture des deux entiers\n");
9         exit(EXIT_FAILURE);
10    }
11
12    int z = 1;
13    if (x > 0 && y < 2) {
14        x = x + 1;
15        y = 0;
16    } else {
17        y = 2 * y;
18        x = 0;
19        z = 0;
20    }
21    printf("%d %d %d\n", x, y, z);
22
23    return EXIT_SUCCESS;
24 }
```

tp4-ex1.c

Quelles sont les valeurs affichées par ce programme lorsque l'utilisateur entre les valeurs suivantes :

- 1) 1 et 1;
- 2) -1 et -2?

Exercice 2

- 1) Écrivez une fonction `int max_if(int x, int y)` qui renvoie le maximum des deux valeurs passées en paramètre. Contrainte : vous devrez utiliser une instruction `if ... else ...`.
- 2) Écrivez une version `int max_op(int x, int y)` de cette fonction qui utilise un opérateur `...?...:...`.
- 3) Testez les dans un programme qui calcule et affiche le maximum de trois valeurs saisies par l'utilisateur

Exercice 3

Une compagnie de locations de voitures propose à ses clients deux tarifs au choix :

- 1) le premier est forfaitaire à la journée dans la limite de 500 kilomètres par jour : le prix est de 80 euros la journée. Si le client dépasse les 500 kilomètres par jour en moyenne, les kilomètres supplémentaires sont facturés au prix p_1 (prix au kilomètre fixé);

- 2) le second tarif est un tarif à tranches. Les 2 000 premiers kilomètres sont facturés à un prix p_2 (prix au kilomètre fixé). Sur les kilomètres suivants, une remise de 11,5 % est accordée au client.

Écrivez un programme prenant en entrée un prix p_1 , un prix p_2 , un nombre de kilomètres et un nombre de jours, et affichant quel est le tarif le plus avantageux pour le client ainsi que la somme qu'il devra payer.

Exercice 4

- 1) Écrire une fonction `void add_1s(int *ph, int *pm, int *ps)` prenant en paramètre les adresses de trois entiers représentant une durée sous la forme heure : minute : seconde et qui modifie ces valeurs de façon à ce qu'une seconde soit ajoutée à la durée représentée. Contrainte : vous ne devrez jamais convertir la durée en seconde et vous devrez utiliser des `if`.
- 2) Écrire une fonction `void sub_1s(int *ph, int *pm, int *ps)` prenant en paramètre les adresses de trois entiers représentant une durée sous la forme heure : minute : seconde et qui modifie ces valeurs de façon à ce qu'une seconde soit retranchée à la durée représentée. Contrainte : vous ne devrez jamais convertir la durée en seconde et vous devrez utiliser des `if`.
- 3) Testez ces fonctions dans un programme. Si vous avez écrit la bibliothèque `timeio.h` au tp précédent, ajoutez lui ces fonctions.

Exercice 5

Écrire un programme complet permettant de résoudre une équation du second degré dont les coefficients sont des réels saisis par l'utilisateur.

Attention : bien prévoir tous les cas possibles, y compris les solutions complexes et dégénérées.