Travaux pratiques n°3 (2h): Les alternatives en langage C

Objectifs

- Programmer des alternatives en langage C

Documents nécessaires

- sujet du TD3 et notes de TD3
- support de cours

Partie 1: Introduction pratique au langage C

Les alternatives en C:

1) Le if ... else:

Le cours présente comment traduire en C les alternatives étudiées en algorithmique. La syntaxe générale est la suivante:

```
if (conditions)
        {
         instruction;
        ...
        }
else
        {
        instruction;
        ...
        }
/* fin du if */
```

Quelques remarques:

- Le **else** est facultatif: S'il n'y a rien a faire dans le cas du else on peut l'omettre.
- S'il y a qu'une seule instruction dans le **then** ou dans le **else** on peut omettre les {} (mais çà marche quand même si on les laisse!)

- Respectez bien l'indentation, elle est indispensable pour la lecture de vos algorithmes (et programmes)

exemple:

```
if (température<0)
        printf ("il gèle\n");
/* fin du if */
```

Question 1:

A titre d'exemple écrire un programme en C correspondant à la traduction du corps de l'algorithme suivant sachant que situ est un variable de type char:

```
if (situ == 'b')
    afficher("Votre situation : Bleu");
 else
    if (situ == 'v')
        afficher("Votre situation : Vert");
    else
        if (situ == 'o')
            afficher("Votre situation : Orange");
        else
            if (situ == 'r')
                afficher("Votre situation : Rouge");
            else
                afficher("Votre situation : Refusé");
```

2) Le switch ... case:

Dans le cas où vous avez une suite d'alternatives qui testent la valeur d'une même expression, vous pouvez les organiser dans un switch ... case qui rendra votre programme beaucoup plus lisible:

La syntaxe est:

```
switch (expression)
                                        {
    case val-expression-1 :
            instruction;
            ...
            break;
    case val-expression-2 :
```

```
                    instruction;
                    ...
                    break;
                ...
               default :
                 instruction;
                                                        ...
}
/* fin du switch */
```

expression est évaluée et fournit une valeur.
Si cette valeur est une des valeurs *val-expression-1* ***val-expression-2,*** ... alors l'exécution se poursuit par les instructions du **case** correspondant et ce jusqu'à la prochaine instruction **`break;`**.

Si **val** n'est pas trouvé et que **default** est présent, alors les instructions suivant **default:** sont exécutées.

val-expression doit être une constante de type entier ou caractère, et doit être unique, dans la suite: ***val-expression-1***, ***val-expression-2***, ...

Question 2:

A titre d'exemple traduire l'algorithme détaillé précédent en utilisant un **switch ... case**.

Partie 2: Traduire des corps d'algorithmes détaillés avec alternatives en langage C

Exercice 1

1) Écrire le programme C correspondant à l'algorithme l'exercice 6 du TD3
2) Écrire le programme C correspondant à l'algorithme de l'exercice 7 du TD3

Exercice 2

Écrire un programme qui propose à l'utilisateur de faire une opération (+,-,*,/) entre deux opérandes et qui affiche le résultat.