LANGAGE C – TP 2
Structures de contrôle - Booléens

Les codes proposés devront impérativement avoir la structure définie dans la fiche de TP1 et des commentaires pertinents seront ajoutés. Une grande importance doit être apportée à la lisibilité des codes.

Exercice 0

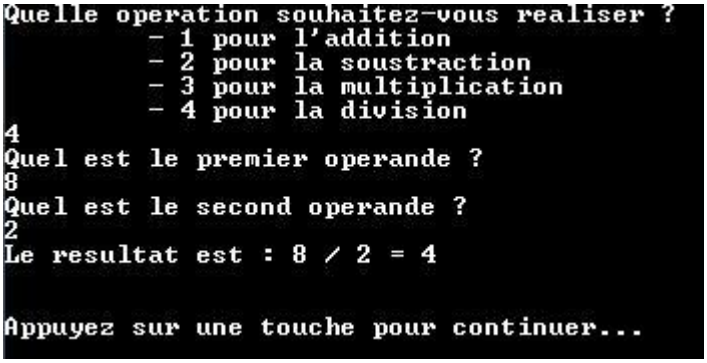
Ecrire un programme qui affiche de manière lisible et détaillée les multiples de 9, jusqu'à 20x9.

Exercice 1

On souhaite écrire un programme qui demande successivement à un utilisateur un opérateur (on se limitera aux cas : +, -, * et /) et deux opérandes et qui affiche le résultat de l'opération demandée. Cette calculatrice sommaire sera réalisée en utilisant un switch. Par ailleurs, un test doit être prévu pour éviter les divisions par 0.

Le type de l'opérande sera associé à une variable de type entier.

Le rendu à l'écran devra impérativement être le suivant :



Remarque : l'utilisation d'un code numérique pour définir les opérations à réaliser s'impose car, si l'on demande à l'utilisateur d'entrer directement '+' et qu'on le stocke dans une variable de type char, le fait de taper « entrer » est codé par « \n » en mémoire, et en attente de stockage dans une variable. Cela pose donc un problème pour la troisième question, à moins de vider le buffer entre temps.

1) Ecrire le code correspondant.
2) Proposer un jeu de tests satisfaisant de manière à valider ce code.
3) Modifier le code pour permettre à l'utilisateur de réaliser plusieurs opérations successives, en lui demandant s'il souhaite réaliser une nouvelle opération. La boucle mise en place devra s'exécuter au moins une fois. Afin de ne pas avoir à gérer les chaînes de caractères en C (objet d'un TP ultérieur, la réponse attendu sera un entier : 1 pour une réponse affirmative). Une variable booléenne sera mise en place, de valeur true si la réponse de l'utilisateur vaut 1. En effet, il n'existe pas de format associé au booléen en C.

Le rendu attendu à l'écran est donc la suivant :

```
Quelle operation souhaitez-vous realiser ?
        - 1 pour l'addition
        - 2 pour la soustraction
        - 3 pour la multiplication
        - 4 pour la division
4
Quel est le premier operande ?
8
Quel est le second operande ?
2
Le resultat est : 8 / 2 = 4

Souhaitez-vous realiser une nouvelle operation (1 pour oui) ?
1

Quelle operation souhaitez-vous realiser ?
        - 1 pour l'addition
        - 2 pour la soustraction
        - 3 pour la multiplication
        - 4 pour la division
3
Quel est le premier operande ?
4
Quel est le second operande ?
5
Le resultat est : 4 * 5 = 20

Souhaitez-vous realiser une nouvelle operation (1 pour oui) ?
0



Appuyez sur une touche pour continuer...
```

Exercice 2

Ecrire un code qui calcule et affiche la somme des n premiers termes de la série harmonique, c'est-à-dire la somme $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$.

La valeur de n sera fournie par l'utilisateur et un test devra être réalisé afin de s'assurer que cet entier est non nul.

Exercice 3

En utilisant la structure de test if/else if/else, écrire un programme affichant par ordre croissant trois entiers donnés par un utilisateur (demander les entiers un par un si le scanf n'accepte pas les 3 entiers en une fois).

Proposer un jeu de tests couvrant toutes les possibilités d'ordre des trois variables afin de valider le programme.

Exercice 4

Programmer le jeu du plus ou moins. Les étapes du jeu sont les suivantes :

1. L'ordinateur tire au sort un nombre entre 1 et 100.
   Les lignes de commande permettant de le faire sont les suivantes :
   - Dans les commandes pré-processeur :

```
# include <time.h>
# include <stdlib.h>
```

- Dans le main() :

```
int mini = 0, maxi = 100;
int nb_mystere;
srand(time(NULL)); // cette commande ne doit être exécutée qu'une seule fois
nb_mystere = (rand() % (maxi - mini + 1)) + mini;
```

2. L'ordinateur demande à l'utilisateur un nombre entre 0 et 100.
3. L'ordinateur compare le nombre entré au nombre mystère et indique si le nombre proposé est plus grand ou plus petit que le nombre mystère. Ou bien signale que l'utilisateur a gagné, si le nombre proposé est identique au nombre mystère.
4. SI le nombre proposé est différent du nombre mystère et que le nombre de proposition de l'utilisateur est inférieur à 10, l'ordinateur demande un nouveau nombre à l'utilisateur.
5. La partie s'arrête soit lorsque l'utilisateur a trouvé le nombre, auquel cas, un message de félicitations s'affiche, ou bien lorsque l'utilisateur a proposé 10 nombres sans trouver le nombre mystère, auquel cas, il a perdu et l'ordinateur lui indique le nombre qu'il fallait trouver.
6. L'ordinateur demande à l'utilisateur s'il souhaite faire une autre partie.