

Cahier de TP n°2

I Boucles et aiguillage

1. Tests sur les caractères

Ecrire un programme C qui lit un caractère entré au clavier puis affiche s'il s'agit d'une lettre majuscule, d'une lettre minuscule, d'un chiffre ou bien d'un caractère non alphanumérique.

2. Affichage de caractères et de leur code ASCII

Ecrire un programme qui affiche les caractères de 'A' jusqu'à 'Z' et de '0' jusqu'à '9' suivis la valeur de leur code ASCII.

Exemple : caractere A code 65

3. Arrêt d'une saisie grâce à une valeur

Ecrire un programme C qui lit une suite de nombres entiers positifs terminée par un entier négatif et qui affiche la moyenne de ces nombres positifs.

4. Recherche d'un nombre caché

Ecrire un programme qui demande à l'utilisateur de trouver un nombre entier secret compris entre 0 et 1000. Pour cela il est guidé par le programme qui lui indique à chaque essai si la valeur saisie est supérieure ou inférieure au nombre secret. L'utilisateur doit trouver le nombre secret en au plus dix essais.

Pour le moment, vous entrerez vous même la valeur du nombre secret, vous verrez plus tard comment obtenir un nombre aléatoire.

Lorsque le jeu se termine, le programme affiche le résultat (gagne en ... essais ou perdu).

En cas de gain, le nombre d'essais est affiché.

5. Menu

- Ecrire un programme proposant une série de choix à l'utilisateur. Le programme affiche uniquement le choix de l'utilisateur et un message d'erreur si le choix n'appartient pas à l'ensemble des choix possibles. Le menu doit être réaffiché après chaque choix tant que le choix d'arrêt n'a pas été sélectionné.

- A l'aide de structures de blocs intégrer les programmes précédents dans votre menu.

(une méthode plus élégante serait l'utilisation de fonctions , cela sera vu très prochainement)

Exemple de menu :

```
#include <stdio.h>

void main(void)
/*Calcullette*/
{
    float A, B, resul;
    char car ;
    scanf("%f%c%f",&A,&car,&B);
    switch(car)
    {
        case '+':resul=A+B;break;
        case '-':resul=A-B;break;
        case '*':resul=A*B;break;
        case '/':resul=A/B;break;
        default :printf("%c n'est pas un operateur\n",car); break;
    }
    printf("%f %c %f=%f\n",A,car,B,resul);
}
```

Intégrer ensuite les instructions de ce programme dans une structure de boucle afin de demander à l'utilisateur s'il veut recommencer. Les réponses acceptées sont 'o' et 'n'.

Attention lorsqu'on utilise scanf pour la lecture d'un caractère, le format "%c" supprime la notion de séparateurs. Tous les symboles sont pris en compte et en particulier le symbole de passage à la ligne. Il faut utiliser le format " %c" pour que les séparateurs ne soient pas mémorisés comme valeur des variables.

II Fonctions

1. Menu

Ecrire un menu utilisant des fonctions pour :

- la moyenne d'une suite d'entiers positifs,
- le test sur les caractères,
- la recherche d'un nombre,
- l'affichage des codes ASCII.

2. Lecture contrôlée d'un entier positif

Ecrire une fonction LirePositif qui effectue la lecture contrôlée d'un entier positif.

Cette fonction demande un entier positif à l'utilisateur tant que celui-ci fournit des entiers négatifs, et renvoie la valeur du premier entier positif saisi.

3. Exponentielle

Ecrire une fonction Exponentielle qui prend en argument deux entiers n et m avec $m \geq 0$ et qui renvoie n^m .

4. Plus grand diviseur commun

Ecrire une fonction Pgcd qui prend en argument deux entiers n et m avec n non nul et qui renvoie le plus grand diviseur commun de |n| et |m|. On utilisera la méthode d'Euclide.

5. Plus petit multiple commun

Ecrire une fonction Ppcm qui prend en arguments deux entiers non nuls n et m et qui renvoie le plus petit multiple commun de n et m.

6. Calculatrice

Ecrire un programme permettant d'utiliser les fonctions précédentes. On ajoutera de plus les opérations de base : addition, soustraction, multiplication, division.

Les opérations seront rentrées sous la forme : opérateur opérande1 opérande2

où opérateur est de type char, opérande1 et opérande2 sont de type int.

On utilisera les symboles E, P, p pour Exponentielle, Plus grand diviseur commun, Plus petit multiple commun. Le résultat de l'opération sera affiché à l'écran.

7. Somme

On définit la somme $S_m(n)$ par : $S_m(n) = 1^m + 2^m + \dots + n^m$ pour $n, m > 0$.

Ecrire une fonction Somme qui prend en argument deux entiers strictement positifs et qui renvoie $S_m(n)$. Ajouter cette fonction à votre calculatrice.

III Fonctions transformant la valeur de paramètres

1. Echange de deux valeurs

Tester les différentes valeurs prises par les paramètres d'appels de la fonction d'échange des valeurs de 2 entiers, en utilisant les modes de transmissions suivant :

- Passage par valeur des deux arguments,
- Passage par valeur des adresses des deux arguments (en utilisant les symboles & et *)

2. Somme de deux fractions

Ecrire une fonction SommeFrac qui reçoit quatre paramètres représentant le numérateur et le dénominateur de deux fractions et qui affecte à deux autres paramètres le numérateur et le dénominateur de la fraction représentant la somme des deux fractions initiales.

3. Transformation des minuscules en majuscules

- Ecrire une fonction TransformeMinMaj qui reçoit un argument représentant une valeur de type char et qui renvoie 1 si cette valeur est une lettre de l'alphabet (minuscule ou majuscule). La fonction doit mettre la variable reçue en majuscule si celle-ci 'était en minuscule.
- Ecrire un programme qui lit une suite de caractères sur une ligne et qui affiche uniquement les caractères alphabétiques de la suite, en majuscule.

Par exemple, si l'utilisateur entre la ligne a4H3n, le programme doit imprimer AHN.