Université Paris-Nord — D.E.U.G. MIAS 1

Institut Galilée — Année 2002-2003

TD 9 : Les tableaux en C.

Exercice 1. (Utilisation du switch)

En utilisant un **switch**, crire un programme C qui demande l'utilisateur d'indiquer s'il parle franais, anglais, espagnol ou une autre langue et rpond:

"Bonjour !" s'il parle franais,
"Hello !" s'il parle anglais,
"Ol !" s'il parle espagnol,
"Dsol !" s'il ne parle, ni franais, ni anglais, ni espagnol.

Indiaction: On pourra demander l'utilisateur de rpondre par `F` s'il parle franais, `A` s'il parle anglais, `E` s'il parle espagnol, ou par tout autre caractre s'il parle une autre langue.

```
#include <stdio.h>
int main()
{
        char i;
        printf("Quelle langue parlez vous ?: Rpondez par\n");
        printf("        F  si c'est en franais\n");
        printf("        A  si c'est en anglais\n");
        printf("        E  si c'est en espagnol\n");
        printf("        Un autre caractre sinon\n");
        scanf("%c",&i);

        switch(i) {
            case 'F': printf("\tBonjour !\n"); break;
            case 'A': printf("\tHello !\n"); break;
            case 'E': printf("\tol !\n"); break;
            default:  printf("\tDsol !\n");
        }
        return(1);
}
```

Exercice 2. (Petite calculatrice)

On veut crire un petit programme C qui arrive faire des oprations du type: $x \triangle y$, o $\triangle$ est une opration arithmtique lmentaire, i.e.: $\triangle \in \{+, -, *, /\}$.
Les oprandes x et y ainsi que l'opration sont introduits au clavier de la manire suivante:

`A` pour l'Addition, `S` pour la Soustraction,
`M` pour la Multiplication et `D` pour la Division.

Solution:
Pour certains compilateurs C, Visual C++ et gcc par exemple, le **scanf**("%c",&charactre); pose des problmes. Il y a plusieurs solutions: (inclure `#include<string.h>`)

- soit utiliser un **fflush**,
- soit mettre un double **getchar();**
- soit utiliser un **scanf**("%s",&charactre);

Il me semble que la solution la moins mauvaise est la troisime.
De toute faon, pour des raisons pdagogiques, je propose de ne pas en parler en TD, mais d'voquer ce problme, au besoin, uniquement en TP. Moyennant ce petit problme de saisie de caractre, la programme C correspondant s'crit:

```
#include <stdio.h>
#include <string.h>
int main()
{
        char op;   /*  Charactre de l'opration  effectuer  */
        float x, y, resultat;
        printf("Donner le reel x :\n");
        scanf("%f", &x);
        printf("Donner le reel y :\n");
        scanf("%f", &y);
        printf("\tA: Addition; S: Soustraction :\n");
        printf("\tM: Multiplication et  D: Division :\n");

        /* ************************************************ */
        /*    ATTENTION si pour un caractere, le scanf ne      */
        /*    fonctionne pas avec %c, c'est le cas de notre    */
        /*    compilateur Visual C++, et de gcc,               */
        /*          utiliser alors  %s.                        */
        /* ************************************************ */

        printf("Donner le type d'operation :\n");
        scanf("%s",&op);

        /*       Calcul de x op y       */
        switch(op) {
            case 'A' :
                 resultat = x+y;
                 printf("\t %f + %f = %f \n", x,y,resultat);
                 break;
            case 'S' :
                 resultat = x-y;
                 printf("\t %f - %f = %f \n", x,y,resultat);
                 break;
            case 'M' :
                 resultat = x*y;
                 printf("\t %f * %f = %f \n", x,y,resultat);
                 break;
            case 'D' :
                 resultat = x/y;
                 printf("\t %f / %f = %f \n", x,y,resultat);
                 break;
            default :  printf("Operation inconnue\n");
        }
        return(1);
}
```

Exercice 3. (Petites manipulations d'un tableau)

Ecrire un programme C qui compte le nombre d'lment non nul d'un tableau d'entiers, affiche ce nombre ainsi que la somme de tous ses lments. La taille N de ce tableau sera dclare comme constante au dpart.

```
#include<stdio.h>
#define N 5
main() {
        int tab[N];
        int i, nombre, somme;
        /*   nombre: nombre d'elements non nul  */

        /*  Saisie des elements du tableau      */
        for(i=0;i<N;i=i+1) {
            printf("Donnez l'element de rang %d:\n",i);
            scanf("%d",&tab[i]);
        }
        nombre=N;
        somme=0;

        /*  Calcul de nombre et la somme     */
        for(i=0;i<N;i=i+1) {
             if(tab[i]!=0) {
                  somme=somme + tab[i];
                  nombre=nombre-1;
             }
        }

        /*  Affichage des resultats  */
        printf("Nombre d'elements nuls = %d et somme =%d\n",nombre,somme);
}
```

Exercice 4. (Programme inconnu)

1. Que fait le programme C suivant ?
2. Rectifier ce programme pour qu'il inverse tous les lments du tableau.

```
#include<stdio.h>
#define N 5
main() {
        int t[N];
        int i, Max, temp;

        for(i=0;i<N;i=i+1) {
            printf("Donnez l'element de rang %d:\n",i);
            scanf("%d",&t[i]);
        }

        printf("Tableau AVANT:\n");
        for(i=0;i<N;i=i+1) printf("t[%d] = %d:\n",i, t[i]);
```

```
        for(i=0;i<N;i=i+1) {
             temp=t[i];
             t[i]= t[N-1-i];
             t[N-1-i]= temp;
        }
        printf("Tableau APRES:\n");
        for(i=0;i<N;i=i+1) printf("t[%d] = %d:\n",i, t[i]);
}
```

Rponse: Alors qu'on a l'impression qu'il va inverser les lments du tableau, ce programme laisse le tableau intacte !
Exemple:

```
Tableau AVANT:
t[0] = 4:
t[1] = -8:
t[2] = 22:
t[3] = 0:
t[4] = 9:
Tableau APRES:
t[0] = 4:
t[1] = -8:
t[2] = 22:
t[3] = 0:
t[4] = 9:
```

Le programme correct pour inverser tous les lments du tableau consiste rduire le champ de la boucle jusqu'au milieu, i.e.: $0 \leq i \leq N/2$, donc remplacer la seconde boucle **for** par

```
for(i=0;i<(N/2);i=i+1) {
    temp=t[i];
    t[i]= t[N-1-i];
    t[N-1-i]= temp;
}
```