

Informatique S1 – Programmation C

Exercices – Corrigés

TD 7 : switch

Dans ce TD, nous allons réaliser des exercices couvrant l'usage de l'instruction switch et de l'opérateur « ? : » dans le langage C.

Objectifs : présentation des tests « switch »

- Présentation des tests switch
 - Usage du break
- Opérateur de test ? :

Exercice 1

```
#include <stdio.h>

/* TD 7 : choix de menu */

int main () {
    int choix;

    /* affichage du menu */
    printf ("Indiquer votre preference :\n");
    printf ("I=====I\n");
    printf ("I [1] - Menu salade          I\n");
    printf ("I [2] - Menu sand. + salade    I\n");
    printf ("I [3] - Menu burger           I\n");
    printf ("I [4] - Menu burger frites     I\n");
    printf ("I [5] - Aucun des precedents   I\n");
    printf ("I=====I\n");
    printf ("\n");

    /* demande le choix */
    printf ("Choix : ");
    scanf ("%d",&choix);

    switch (choix) {
        case 1:
            printf ("Choix : Menu salade \n");
            printf ("          Bravo ! Votre sante vous remercie.\n");
            break;

        case 2:
            printf ("Choix : Menu sans + salade \n");
            printf ("          C'est bien de faire attention a votre
ligne.\n");
            break;

        case 3:
            printf ("Choix : Menu burger \n");
            printf ("          Attention aux repas non-equilibres\n");
            break;

        case 4:
            printf ("Choix : Menu burger frites \n");
            printf ("          Attention a votre cholestérol.\n");
```

```
        break;
    case 5:
        printf ("Choix : Aucun des precedents \n");
        printf ("                N'oubliez pas qu'il faut manger
equilibre.\n");
        break;
    default:
        printf ("Choix invalide! \n");
    }
}
```

Exercice 2

Note : les étudiants n'ont pas vu en cours les fonctions « toupper » et « tolower » (bibliothèque `ctype.h`), ni les fonctions de type « isdigit », « isalpha », etc. (aussi dans la bibliothèque `ctype.h`). Les premières peuvent être expliquées lors du TD, s'il y a l'intérêt de la part des étudiants. Les dernières peuvent être également expliquées, mais l'exercice demande de manière explicite l'usage de l'instruction `switch`.

```
#include <stdio.h>
/* TD 7 : switch voyelle/chiffre/consonne */
int main () {
    char c;

    printf ("Entrez un caractere : ");
    scanf ("%c", &c);

    switch (c) {
        case 'a': case 'A':
        case 'e': case 'E':
        case 'i': case 'I':
        case 'o': case 'O':
        case 'u': case 'U':
            printf ("voyelle \n");
            break;

        case '0':
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
            printf ("chiffre \n");
            break;

        case 'b': case 'B':
        case 'c': case 'C':
        case 'd': case 'D':
        case 'f': case 'F':
        case 'g': case 'G':
        case 'h': case 'H':
        case 'j': case 'J':
        case 'k': case 'K':
        case 'l': case 'L':
        case 'm': case 'M':
        case 'n': case 'N':
```

```
case 'p': case 'P':
case 'q': case 'Q':
case 'r': case 'R':
case 's': case 'S':
case 't': case 'T':
case 'v': case 'V':
case 'x': case 'X':
case 'z': case 'Z':
    printf ("consonne \n");
    break;
default:
    printf ("%c est inconnu (caractere de controle?) \n",c );
}
}
```

Exercice 3 :

```
#include <stdio.h>

/*  TD 7 : calculette  */

int main () {
    float a, b, resultat;
    char op;

    //entre des donnees
    printf ("Entrez l'operation [+ - * /]: ");
    op = getchar();
    printf ("Entrez a : ");
    scanf ("%f", &a);
    printf ("Entrez b : ");
    scanf ("%f", &b);

    resultat = 0.0;
    //calculette
    switch (op) {
        case '+':
            resultat = a + b;
            break;
        case '-':
            resultat = a - b;
            break;
        case '*':
            resultat = a * b;
            break;
        case '/':
            resultat = a / b;
            break;
        default :
            printf ("Operateur %c inconnu !\n", op);
    }

    printf ("%f %c %f = %f \n", a, op, b, resultat);
}
```

Exercice 4

```
#include <stdio.h>

/* TD 7 : operateur ? : */

int main () {
    int a, b;

    printf ("Entrez a : ");
    scanf ("%d", &a);
    printf ("Entrez b : ");
    scanf ("%d", &b);

    printf ("a divise b : %c \n",
            (b%a == 0) ? 'O' : 'N');
    printf ("b divise a : %c \n",
            (a%b == 0) ? 'O' : 'N');
    printf ("a est impair : %c \n",
            (a%2 != 0) ? 'O' : 'N');
    printf ("b est impair : %c \n",
            (b%2 != 0) ? 'O' : 'N');
    printf ("a < b : %c \n",
            (a<b) ? 'O' : 'N');
    printf ("a > b : %c \n",
            (a>b) ? 'O' : 'N');
}
```

Exercice 5

En cours, ils ont eu un exemple de devinette qui calculait un numéro aléatoire que l'utilisateur devrait trouver. Cet exercice est donc une extension de l'exemple vu en cours, ce qui n'enlève en rien sa complexité. Ils auront probablement des difficultés pour terminer la boucle avant le troisième essai, si l'utilisateur trouve la bonne réponse avant celui-ci (éviter d'utiliser break pour cela), et aussi pour pouvoir montrer la réponse uniquement si l'utilisateur ne l'a pas trouvé.

Note : les étudiant n'ont pas vu « define » en cours (donc, pas de #define FALSE 0).

```
#include <stdio.h>
#include <stdlib.h> //fonction rand
#include <time.h> //fonction time

/* cm7 : swith (devinette) */
int main ()
{
    int essai,x;
    int i;
    int trouve = 0;

    //numero mystere
    srand (time(NULL)); //initialise le generateur
    x = rand()%11;

    //3 essais ou jusqu'a qu'on trouve la reponse
    for (essai=1; essai<=3 && !trouve; essai++) {
        printf ("Essai %d: \nEntrez i [0-10] : ", essai);
        scanf ("%d", &i);

        switch (x-i) {
```

```
        case 0:
            printf ("bingo!!\n");
            trouve = 1; //pour terminer
            break;
        case 1:
        case -1:
            printf ("presque\n");
            break;
        case 2:
        case -2:
        case 3:
        case -3:
            printf ("proche\n");
            break;
        default:
            printf ("loin\n");
    }
}

//si on n'a pas trouve la reponse, on la montre
if (!trouve)
    printf ("Damage! Reponse : %d\n",x);
}
```