

## Principes et méthodes de programmation

Exercices du 29 octobre 2009 (semaine 9)

1) Dans un pays les taux de taxation sur les revenus sont les suivants :

- de 1€ à 10000€ : 4% de taxe sur le revenu
- de 10001€ à 20000€ : 14% de taxes sur le revenu
- de 20001€ à 30000€ : 26% de taxes sur le revenu
- de 30001€ à 40000€ : 32% de taxes sur le revenu

Faire un programme qui quand on introduit le revenu affiche la taxation.

### code

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    int i;

    printf(" Entrez une valeur : ");
    scanf("%d", &i);

    if(i>0 && i<=10000) printf("\n --> %d * 4%% : %d \n\n", i, i*.04);
    else if(i<=20000) printf("\n --> %d * 14%% : %d \n\n", i, i*.14);
    else if(i<=30000) printf("\n --> %d * 26%% : %d \n\n", i, i*.26);
    else if(i<=40000) printf("\n --> %d * 32%% : %d \n\n", i, i*.32);
    else printf("\n Montant incorrect !!! \n\n");

    return 0;
}
```

2) Écrire un programme permettant d'ordonner des valeurs de type caractère dans un tableau.

Soit, à l'écran :

```
-- C -- | -- A --
-- O -- | -- C --
-- S -- | -- F --
-- F -- | -- G --
-- J -- | -- J --
-- U -- | -- M --
-- G -- | -- O --
-- M -- | -- S --
-- Z -- | -- U --
-- A -- | -- Z --
```

code permettant le tri :

```
while(cpt1++<NBR_LETTRE) {
    cpt2=cpt1;
    while(cpt2++<NBR_LETTRE)
        if(tabOrd[cpt1-1]>=tabOrd[cpt2-1]) {
            aux=tabOrd[cpt1-1];
            tabOrd[cpt1-1]=tabOrd[cpt2-1];
            tabOrd[cpt2-1]=aux;
        }
}
```

*NBR\_LETTRE* : nombre de cellule du tableau

*cpt1, cpt2* : compteur de type entier

*tab* : tableau contenant les lettres non triées

*tabOrd* : tableau trié selon l'ordre alphabétique croissant

*aux* : valeur intermédiaire utilisée pour permuter les positions dans le tableau

## code

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

#define NBR_LETTRE 10

void init(char *tab);
void copie(char *tab, char *tabOrd);
void afficher(char *tab, char *tabOrd);

int main(int argc, char *argv[]) {

    char tab[NBR_LETTRE], tabOrd[NBR_LETTRE], aux;
    int cpt1=0, cpt2;

    init(tab);
    copie(tab, tabOrd);

    while(cpt1++<NBR_LETTRE) {
        cpt2=cpt1;
        while(cpt2++<NBR_LETTRE)
            if(tabOrd[cpt1-1]>=tabOrd[cpt2-1]) {
                aux=tabOrd[cpt1-1];
                tabOrd[cpt1-1]=tabOrd[cpt2-1];
                tabOrd[cpt2-1]=aux;
            }
        afficher(tab, tabOrd);
        return 0;
    }
}

void init(char *tab) {

    int i, j, unique;
    srand((unsigned)time(NULL));

    for(i=0; i<NBR_LETTRE; i++) {
        tab[i]=(char)(rand()%26+65);
        if(i) {
            unique=1;
            for(j=0; j<i; j++)
                if(tab[j]==tab[i]) unique=0;
            if(!unique) i--;
        }
    }
}

void copie(char *tab, char *tabOrd) {

    int i;

    for(i=0; i<NBR_LETTRE; i++) tabOrd[i]=tab[i];
}

void afficher(char *tab, char *tabOrd) {

    int i=0;

    while(i++<NBR_LETTRE) printf("-- %c -- | -- %c --\n", tab[i-1], tabOrd[i-1]);
}
```