Programmation 2

L1 2011-2012

Travaux Pratiques 1 - Rappels

On rappelle que un fichier de code source c se termine par .c. La commande de compilation est `gcc fichier_source.c`. Le nom du fichier de sortie est par défaut `a.out`. Les options de la commande de compilation sont :

- `-o nom_sortie` pour donner un nom au fichier de sortie.
- `-Wall` pour demander au compilateur d'afficher un maximum d'information.
- `-ansi` pour compiler selon la norme ansi.

Ces trois options doivent être appliqués. Il est très fortement déconseillé d'utiliser des variables globales.

Exercice 1.

Écrire et exécuter le code suivant.

```
#include<stdio.h>

int main(int argc, char** argv)
{
        int i=1;
        int* p;

        printf("Valeur de i : %d\n",i);
        printf("Addresse de i : %p\n",&i);

        p=&i;
        printf("instruction : p=&i\n");
        printf("Valeur de p : %p\n",p);

        printf("instruction : $*p=0\n");
        *p=0;

        printf("Valeur de i : %d\n",i);
        printf("Valeur de p : %p\n",p);
        return 0;
}
```

Que représente l'expression `&a` ? Que représente l'expression `*p` ?

Exercice 2.

1. Écrire une fonction `initialiser` qui prend en paramètre l'adresse d'un `int` et qui initialise cet entier à 1.
2. Écrire une fonction `afficher_adresse_entier` qui prend en paramètre l' adresse d'un `int` et qui affiche la valeur de cet entier et son adresse.
3. Écrire une fonction `echange` qui prend en paramètre deux adresse d'entier et qui echange leur valeurs.

Exercice 3. Nombre factoriels

On dit qu'un entier strictement positif x est un nombre factoriel s'il existe un autre entier `y` tel que $x = y!$.

Écrire une fonction qui, étant donné un entier x, vérifie si x est un nombre factoriel et, si c'est le cas, calcule la valeur de y tel que $x = y!$. En déduire un programme complet.

Exercice 4. Nombres parfaits

Un nombre parfait est un entier positif qui est égal à la somme de tous ses diviseurs, excepté lui-même. Par exemple, 6 est un nombre parfait ($6 = 1 + 2 + 3$).

1. Écrire une fonction `est_parfait` qui prend en argument un entier positif `n` et qui retourne `1` si `n` est parfait et `0` dans le cas contraire.
2. Écrire une fonction qui affiche tous les nombres parfaits inférieurs à un entier `n`.
3. Écrire une autre fonction qui affiche les `n` premiers nombres parfaits.

Exercice 5. Manipulation de tableaux de caractères

Dans la suite les tableaux peuvent contenir au plus 50 caractères. Écrire les fonctions pour :

1. lire un tableau de caractères ; la fonction renvoie le nombre de caractères lus et arrête la saisie lorsque le tableau est plein ou que l'utilisateur a appuyé la touche ENTRÉE.
2. afficher sur un ligne un tableau de 50 caractères ; l'affichage du tableau sera suivi sur la ligne suivante par celui des codes ASCII des caractères.
3. compter les voyelles dans un tableau de 50 caractères.
4. compter les caractères qui ne sont ni des chiffres ni des lettres dans un tableau de 50 caractère.

5. remplacer dans un tableau de 50 caractères, chaque voyelle en minuscule par la voyelle en majuscule correspondante.

Il est conseillé d'écrire des fonctions additionnelles : `int EstChiffre(char c)` qui renvoie 1 si c est un chiffre, `int EstVoyelle(char c)`, `char MinusMajus(char c)` qui renvoie la majuscule correspondante si c est une minuscule et renvoie c sinon.