

TD 7 : Révisions

Programmation en C (LC4)

Semaine du 12 mars 2007

1 Divers

Exercice 1 Qu'affiche la dernière ligne de la fonction suivante, sachant que la première ligne affichée est :

```
premier : 61, sec : 72, tri : 9, qua : 20
```

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

```
void test(void) {  
    char prem = 'a', sec = 'r', tri = '\t', qua = '\n';  
    unsigned int res;  
    printf("premier:_%x,sec:_%x,tri:_%x,qua:_%x\n", prem, sec, tri, qua);  
    res = (qua << 24) | (tri << 16) | (sec << 8) | prem;  
    printf("res:_%x\n", res);  
}
```

Remarque : le format "%x" permet d'afficher un entier en base hexadécimale.

Exercice 2 Les caractères représentant les lettres de l'alphabet dans le code ASCII ont des valeurs entières successives, c'est à dire que la valeur entière de 'b' correspond à 'a'+ 1.

Écrivez une fonction **void** majuscule(**char** *mot) qui prend en argument une chaîne de caractères en minuscules (on ne vérifiera que c'est bien le cas) et qui la modifie pour qu'elle soit en majuscules.

Écrire l'appel à cette fonction dans main() pour mettre en majuscules les arguments du programme (à l'exception du nom du programme) et les afficher.

Exercice 3 Qu'affiche la fonction suivante ?

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

```
void test(void) {  
    int i = 36;  
    double j = 6.2;  
    printf("%d,_%d\n", i / (int) j, (int) (i / j));  
}
```

Exercice 4

- Écrire une fonction vérifiant si trois nombres entiers a , b , c forment un « triplet de Pythagore », c'est-à-dire si $a^2 + b^2 = c^2$.
- Écrire ensuite un programme permettant d'afficher les N premiers (N entré par l'utilisateur) triplets de Pythagore (a, b, c) tels que $0 < a < b < c$.

Voici ce que devrait afficher un tel programme pour $N = 2$:

```
1: 3^2 + 4^2 = 5^2 (= 25)  
2: 6^2 + 8^2 = 10^2 (= 100)
```

2 Tableaux

Exercice 5 On suppose qu'on manipule des tableaux de double. Écrivez les fonctions réalisant les opérations suivantes :

- renvoyer le plus grand élément d'un tableau
- échanger deux éléments d'un tableau donnés par leur indice
- vérifier qu'un tableau est trié dans l'ordre croissant
- renverser l'ordre des éléments d'un tableau
- renvoyer la concaténation de deux tableaux donnés en argument

3 Pointeurs

Exercice 6 Que valent `i` et `j` à la fin de cette séquence d'instructions ?

```
int i, j, *ptr;
ptr = &i; i = 22; j = *ptr; *ptr = 17;
```

Exercice 7 Dans un programme, on trouve les déclarations et initialisations suivantes :

```
char s1[] = "Un", s2[] = "Deux", s3[] = "Trois";
char *tab[] = { s1, s2, s3};
```

Que désignent les expressions suivantes ?

```
&tab[0], *tab, tab[0], tab[1], *(tab + 1), **tab, *tab[0], *(tab + 1), *tab[1]
```

Exercice 8 On considère la fonction suivante qui permute conditionnellement deux entiers `a` et `b`.

```
#include <stdio.h>

void permute(int a, int b) {
    int t;
    if (a > b) {
        t = a; a = b; b = t;
    }
}

int main(void) {
    int i = 5, j = 2;
    printf("%d_%d\n", i, j);
    permute(i, j);
    printf("%d_%d\n", i, j);
    return 0;
}
```

Corriger le programme pour que `i` et `j` soient réellement permutés.

4 Chaînes de caractères

Exercice 9

- En utilisant la fonction `int getchar(void)` (déclarée dans `<stdio.h>`) qui lit un caractère saisi au clavier et le renvoie (sous forme d'un `int`), écrire une fonction qui lit et renvoie une chaîne de caractères tapée au clavier. Le caractère de saut de ligne `'\n'` indiquera la fin de la saisie.
- Écrire une fonction qui prend en argument une chaîne de caractères et qui renvoie une nouvelle chaîne de caractères contenant l'inversion de la chaîne de caractères donnée en argument.

- On appelle palindrome une suite de caractères qui se lit de la même façon dans les deux sens (exemple : "laval", "ressasser"). Écrire une fonction qui teste si une chaîne est un palindrome.
- Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères et affiche un message pour indiquer si la chaîne est un palindrome.

5 Fonctions récursives

Exercice 10 Écrire une fonction récursive calculant le coefficient binomial $\binom{n}{p}$.

Rappel : $\binom{n}{0} = 1$, $\binom{n}{n} = 1$ et $\binom{n}{p} = \binom{n-1}{p-1} + \binom{n-1}{p}$ pour $p \in \llbracket 1, n-1 \rrbracket$

6 Listes chaînées

Exercice 11 Soit P un polynôme de degré d et de coefficients p_i ($i \in \llbracket 0, d \rrbracket$) : $P[X] = \sum_{i=0}^d p_i X^i$

On souhaite représenter P en occupant le moins d'espace mémoire possible, en particulier quand le degré est élevé et/ou qu'il y a peu de coefficients non nuls.

L'idée consiste à ne considérer que les monômes $p_i X^i$ « utiles » c'est-à-dire tels que $p_i \neq 0$. Les monômes « utiles » seront stockés dans une liste chaînée dont chaque élément contiendra un **double** (le coefficient p_i du monôme) et un **int** (l'exposant i du monôme). Chaque coefficient d'un monôme « utile » sera ainsi chaîné via un pointeur au prochain monôme « utile » de degré strictement inférieur (ou NULL s'il n'y a plus de monôme « utile »).

- Écrire précisément la structure de liste chaînée à utiliser.
- Quelle est l'avantage de cette représentation par rapport à celle utilisant un tableau pour stocker les coefficients du polynôme. Illustrez votre réponse avec l'exemple suivant : $P[X] = 71X^{1000000} - 927X^{1000} + X^{91} - 7X + 3$.
- Écrire une fonction d'affichage.
- Écrire une fonction convertissant un tableau de coefficients en liste de monôme « utiles ».
- Écrire une fonction de dérivation d'un polynôme sous cette forme.
- Écrire une fonction d'addition de deux polynômes sous cette forme.