

Université d'Aix-Marseille III - Faculté des Sciences et Techniques de Saint-Jérôme  
Licence Sciences et Technologies - Mention Maths-Info  
I3 - Algorithmique et Programmation

TD n°2

**Visibilité/portée des variables**

1. On considère le programme C suivant :

```
int a;

int f(int b)
{
    int c;
    c = 5 * b;
    return c;
}

int g(int d)
{
    return d * a;
}

main()
{
    int e;

    a = 3;
    e = 2;
    e = f(7) + g(11);
    printf("%d\n", e);
}
```

- 1.1. Qu'affiche le programme lors de son exécution?
- 1.2. Même question en renommant la variable `e` par `a` dans la fonction `main`.
- 1.3. Même question en renommant la variable `b` par `a` dans la fonction `f`.
- 1.4. Même question en remplaçant `return d * a;` par `return d * e;` dans `g`.
- 1.5. Même question en remplaçant `e = f(7) + g(11);` par `e = f(e) + g(a);` dans la fonction `main`.
- 1.6. Même question en remplaçant la fonction `f` par :

```
int f(int a)
{
    a = g(a) * a;
    return a;
}
```

**Appel de fonction : passage de références**

2. Décomposition d'un entier en carrés.
- 2.1. Ecrire la fonction `soustrait_carre` qui prend en paramètre un entier  $n$  positif passé par référence. Cette fonction retourne la valeur entière  $p$  telle que  $p^2 \leq n < (p+1)^2$  et modifie  $n$  en lui soustrayant la valeur  $p^2$ .

- 2.2.** Ecrire un programme qui affiche la décomposition en carrés d'un entier strictement positif saisi au clavier. On utilisera la fonction `soustrait_carre`. L'affichage se fera sous la forme :
- $$99 = 9^2 + 4^2 + 1^2 + 1^2.$$

**3.** On considère le type `chaîne` défini ainsi :

```
#define MAX ...  
typedef char chaîne [MAX];
```

Si `c` est une variable de type chaîne, `c[i]` représente le  $i+1$  ème caractère de `c`. Pour indiquer la fin de la chaîne, on place, dans la case qui suit le dernier caractère, le caractère spécial `'\0'`. Ainsi, on peut représenter des chaînes de caractères de longueur au plus `MAX-1`.

Par exemple, avec `MAX=10`, la chaîne "bonjour" sera représentée ainsi :

'b'	'o'	'n'	'j'	'o'	'u'	'r'	'\0'		
-----	-----	-----	-----	-----	-----	-----	------	--	--

- 3.1.** Ecrire la procédure `lit_chaine` qui prend en paramètre une variable de type chaîne et la remplit à l'aide de caractères saisis au clavier. On utilisera la procédure prédéfinie `scanf`.
- 3.2.** On considère la fonction prédéfinie `getchar` qui a pour prototype : `char getchar()`. Cette fonction attend qu'on appuie sur une touche du clavier et retourne le caractère correspondant (de type `char`). Réécrire la procédure `lit_chaine` en utilisant la fonction `getchar`.
- 3.3.** Ecrire la fonction `mot_dans_phrase` qui a trois paramètres : une chaîne de caractères phrase contenant un ou plusieurs mots séparés par le caractère espace, un entier `n` et un paramètre de sortie qui contiendra le  $n^{eme}$  mot de phrase s'il existe. La fonction retourne 0 si la phrase possède moins de `n` mots et 1 sinon.
- 3.4.** Ecrire la procédure `nb_mot_car` qui a trois paramètres : une chaîne de caractères phrase (contenant un ou plusieurs mots séparés par le caractère espace) et deux paramètres de sortie : un qui contiendra la longueur de phrase et un autre qui contiendra le nombre de mots de phrase.
- 3.5.** Ecrire un programme utilisant exclusivement les fonctions et procédures précédentes pour lire une phrase au clavier et afficher la longueur de son dernier mot.