Ecole Nationale d'Ingénieurs de Tunis
Année universitaire 2014/2015
Module : Programmation procédurale

TD2 : fonctions et tableaux

Exercice n° 1 (Fonctions qui renvoient ou affichent un résultat.)
Écrire un programme qui définit et utilise :

une fonction somme (int x,int y); qui renvoie la somme de 2 éléments

- une fonction fact(n) qui renvoie la factorielle du nombre n.
- une fonction affiche fact(n) qui ne renvoie rien et affiche la factorielle du nombre n.
- une fonction comb (int n , int p) qui renvoie le nombre de combinaisons de p elements parmi n
- une fonction estDivisible(a, b) qui renvoie 1 si a est divisible par b
- une fonction estPremier(n) qui renvoie 1 si n est premier

Le programme principal permet la saisie d'un entier n et d'un entier p inferieur à n. Ensuite il permet d'afficher la somme de n et p, puis il calcule le nombre de combinaisons de p objets parmi n. Et affiche si ce nombre est premier ou non.

Rappels :

- factorielle(n) = n ! = n (n-1) (n-2) ...1.
- un nombre entier n est dit "premier" s'il n'existe aucun entier d dans l'intervalle [2,n -1] tel que n soit divisible par d.

Exercice n° 2

Ecrire un programme en **langage C** qui lit la dimension N d'un tableau T du type **int** (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier avec une fonction *remplir_tab* et affiche le tableau avec une fonction *affiche_tab*.
Effacer ensuite toutes les occurrences de la valeur 0 dans le tableau T et tasser les éléments restants avec une fonction *efface_zéro*. Afficher le tableau résultant.

Exercice n° 3

Ecrire un programme en **langage C** qui lit les dimensions L et C d'un tableau T à deux dimensions du type **int** (dimensions maximales: 50 lignes et 50 colonnes). Remplir le tableau par des valeurs entrées au clavier avec une fonction *remplir_matrice* et afficher le tableau avec une fonction *affiche_matrice*. Ensuite le programme appel la fonction qui calcule et affiche la somme de chaque ligne et de chaque colonne en n'utilisant qu'une variable d'aide pour la somme.

Exercice n° 4 (Tri par sélection)

Tous les types simples comme les entiers ou les caractères sont munis d'un ordre total permettant de comparer deux éléments. Trier un tableau d'éléments d'un ensemble totalement ordonné consiste à ranger les éléments du tableau de manière croissante par rapport à cet ordre. Il y a plusieurs méthodes de tri, parmi lesquels les tris par sélection et les tris par insertion. Dans ces algorithmes de tri, il est exigé de ne pas utiliser de tableau auxiliaire. Le tableau comportera des entiers positifs avec une valeur négative marquant la fin du tableau.

La méthode de tri par sélection "ordinaire" consiste à sélectionner la position du plus petit élément du tableau, à placer cet élément à la première place par échange puis à recommencer sur le reste du tableau.

1. Écrire d'abord une fonction qui recherche la position du minimum d'une portion d'un tableau (à partir d'une certain indice `courant` jusqu'à un indice `dernier`).
2. Écrire ensuite une fonction de tri par sélection qui trie le tableau donné en paramètre jusqu'à l'indice `taille-1`.
3. Écrire un programme qui lit une taille `taille`, puis les `taille` éléments d'un tableau d'entier à trier, et enfin affiche le tableau trié. La taille maximale du tableau est fixée par une constante `NMAX` dans le programme.

Exercice n° 5 (Recherche d'un élément)

Écrire une fonction search qui recherche une valeur particulière dans un tableau de flottants (float). La fonction prendra en paramètre le tableau, sa taille et la valeur à rechercher. Elle retourna une valeur entière qui sera une position de la valeur dans le tableau si elle est présente dans le tableau et -1 sinon. Inclure cette fonction dans un programme qui saisit les valeurs du tableau et la valeur à rechercher.