IUT Lannion Année 2022-2023

# Département Informatique Semestre 1

# **R1.01 - Initiation au développement**

Contrôle de TP n°2

**Exercice 1** : récursivité et fonction de test

**Question 1**

Écrivez une fonction récursive[[1]](#footnote-0) qui calcule et retourne le terme général Un de la suite définie par :

U0 = 1,

et Un = 2\*Un-1 + 2 pour tout entier n > 0,

*n* étant un entier fourni en paramètre d’entrée de la fonction.

**Question 2**

Écrivez une procédure afficherResultat(…) qui prend deux entiers en paramètres d’entrée : une valeur n et la valeur de Un attendue. La fonction devra afficher la valeur de n, la valeur de Un attendue et la valeur de Un calculée par la fonction de la question 1, comme sur cet exemple :

valeur : 2 résultat attendu : 4 résultat obtenu : 4

**Question 3**

Écrivez un main qui appelle la fonction afficherResultat pour chacun des cas suivants :

n Un attendu

2 10

3 22

4 46

5 94

**Exercice 2** : procédures, fonctions et tableaux 1D

Lors d’une kermesse, une association organise une loterie. Chaque participant doit écrire son nom sur un bulletin et y entourer 5 nombres dans une grille qui en contient 100 (de 0 à 99). En fin de journée, lorsque le tirage a eu lieu, il faut saisir les bulletins et déterminer le ou les gagnant(s).

On appelle *combinaison* une série de 5 nombres compris entre 0 et 99. Elle représente aussi bien le tirage que les 5 nombres entourés sur un bulletin. Le type combinaison est défini comme suit :

#define N 5

typedef int combinaison[N] ;

**Question 1**

Écrivez une **procédure** qui initialise à -1 tous les éléments d’une combinaison passée en paramètre.

**Question 2**

Écrivez une **procédure** qui lit au clavier 5 nombres et les stocke dans une combinaison passée en paramètre. La procédure devra vérifier que les valeurs sont comprises entre 0 et 99. Par contre, on supposera que l’utilisateur ne saisit pas deux fois la même valeur.

**Question 3**

Écrivez une **fonction** qui retourne true si le nombre passé en paramètre est présent dans une combinaison, elle aussi passée en paramètre. La fonction doit retourner false sinon.

**Question 4**

Écrivez une **procédure** qui affiche une combinaison passée en paramètre, sous la forme :

10 – 99 – 5 – 39 - 57

En cas d’erreur (si la combinaison contient -1), la procédure affichera un message.

**Question 5**

Écrivez une **procédure** qui génère aléatoirement 5 nombres entiers compris entre 0 et 99 et qui les stocke dans une combinaison passée en paramètre. On rappelle que la fonction rand() retourne un entier compris entre 0 et RAND\_MAX, et qu’il faut impérativement exécuter srand(time(NULL)); au début du programme.

ATTENTION : la fonction doit générer 5 nombres entiers distincts (on n’admet pas deux fois la même valeur dans le tirage).

**Question 6**

Écrivez une **procédure** qui prend deux combinaisons en paramètres (un bulletin et le tirage) et qui fournit en paramètre de sortie le nombre de "bons" numéros du bulletin, c’est à dire le nombre de numéros du bulletin qui sont présents dans le tirage.

Exemple : si le tirage est 17 - 5 - 78 - 22 - 9 et que le bulletin possède les numéros 19 - 67 - 32 - 5 - 17, alors la fonction doit fournir 2 en paramètre de sortie (en effet, le bulletin possède 2 "bons" numéros).

**Question 7**

Écrivez une **fonction** qui prend une combinaison en paramètre et qui retourne le plus grand nombre que contient cette combinaison. En cas d’erreur (si la combinaison contient -1), la fonction devra retourner -1.

**Question 8**

Écrivez un **main** qui effectue un tirage, puis qui lit au clavier une suite de bulletins avec pour chaque bulletin, la saisie du nom du participant et des 5 nombres entourés. La saisie s’arrête quand on tape \* à la place du nom. Le main affichera à chaque fois le nom du participant et le nombre de bons numéros, comme sur l’exemple suivant.

Exemple d’exécution (les saisies de l’utilisateur sont en gras) :

Nom : **DUCHEMIN**

Bulletin :

**4**

**45**

**56**

**16**

**77**

Resultat DUCHEMIN : 2 bons numeros

Nom : **DUMOULIN**

Bulletin :

**8**

**64**

**32**

**99**

**0**

Resultat DUMOULIN : 3 bons numeros

Nom : **\***

1. Si vous n’arrivez pas à écrire une fonction récursive, fournissez une version itérative de la fonction. [↑](#footnote-ref-0)