Université Abou Bekr Belkaid – Tlemcen
Faculté des Sciences
Département d'Informatique

Année Universitaire 2023-2024
2ème Année - Cycle Ingénieur
Algo & complexité

T R A V A U X P R A T I Q U E S
E V A L U A T I O N

Exercice 1 :

On souhaite vérifier l'égalité de deux chaines de caractères ; écrire une fonction itérative et une autre récursive ainsi qu'un programme principal pour tester vos deux fonctions.

Remarque : Ne pas utiliser de fonctions prédéfinies.

Exercice 2 :

Un **triangle pascal** est un tableau triangulaire de nombres (voir l'exemple) ; Écrire une fonction qui affiche la liste des premières lignes du triangle de Pascal selon le principe suivant :

1) Le sommet est constitué du nombre 1.
2) Chaque ligne commence et finit par 1.
3) Chaque ligne, à partir de la deuxième, possède un élément de plus que la précédente.
4) Chaque autre nombre est la somme des deux nombres : l'un situé juste au-dessus, l'autre situé juste au-dessus à gauche.

Exemple : Nombre de lignes : 10 ↵

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1

Solution Exercice 1

```
#include <stdio.h>
int Strcmp_iterative(const char *str1, const char *str2) {
    while (*str1 != '\0' && *str2 != '\0') {
        if (*str1 != *str2)
            return 0; // Strings are not equal

        str1++;
        str2++;
    }

    if (*str1 == '\0' && *str2 == '\0')
        return 1; // Strings are equal
     else
        return 0; // Strings have different lengths

}

int StrcmpRecursive(const char *str1, const char *str2) {
    if (*str1 == '\0' && *str2 == '\0')
        return 1; // Both strings are empty, so they are equal
     else if (*str1 != *str2)
        return 0; //Characters do not match, the strings are not equal
     else
        // Recursively compare the rest of the strings
        return StrcmpRecursive(str1 + 1, str2 + 1);

     }

int main() {
    char *str1 = "Hello";
    char *str2 = "World";

    if (Strcmp_iterative(str1, str2))
        printf("iterative : The strings are equal.\n");
     else
        printf("iterative : The strings are not equal.\n");


    if (StrcmpRecursive (str1, str2))
        printf("Recursive : The strings are equal.\n");
     else
        printf("Recursive : The strings are not equal.\n");

    return 0;
}
```

Solution Exercice 2

Version 1:

```
#include <stdio.h>

 // Function to calculate and display Pascal's Triangle

    void displayPascalsTriangle(int n) {

        int triangle[100][100];

         triangle[0][0] = 1; // Initialize first row

 // Compute subsequent rows

        for (int i = 1; i< n; i++)

            for (int j = 0; j <= i; j++)

                if (j == 0 || j == i)  triangle[i][j] = 1;

                 else

                    triangle[i][j]=triangle[i-1][j-1]+triangle[i-1][j];


  // Display triangle with proper formatting

    printf("Pascal's Triangle (up to row %d):\n", n);

        for (int i = 0; i< n; i++) {

            for (int j = 0; j <= i; j++)

                      printf("%4d", triangle[i][j]);

            printf("\n");

        }

    }

     int main() {

        int n = 10; // Number of rows to generate

    displayPascalsTriangle(n);

        return 0;

    }
```

Version 2 :

```
#include <stdio.h>

int Pascal(int n, int i) {

    if (i == 0 || n == i)

        return 1;

     else

        return Pascal(n - 1, i) + Pascal(n - 1, i - 1);

    }

int main() {

    int n = 0;

    int k, m;

    printf("Nombre de lignes: ");

    scanf("%d", &n);


    for (k = 0; k < n; k++) {

        for (m = 0; m <= k; m++) {

            int f = Pascal(k, m);

            printf("%d ", f);

        }

        printf("\n");

    }
    return 0;
}
```