

1. Ecrire une fonction double `serie(int n, double x)` qui renvoie la somme $\sum_{i=0}^n u_i$ où

$$u_0 = 1$$
$$u_i = \frac{u_{i-1} * x}{i}, i \geq 1$$

2. Soit le programme suivant.

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

void
f(char s[])
{
    int l= strlen(s);
    if (l != 0)
        s[l-1]='\0';
}

void
g(char s[])
{
    for(int i=0; s[i] != '\0' ; i++)
        s[i]= s[i+1];
}

int
main(void)
{
    char s[]="totototo";
    f(s);
    printf("<%s>\n", s);
    g(s);
    printf("<%s>\n", s);
    return EXIT_SUCCESS;
}
```

- (a) Qu'affiche le programme ?
- (b) Écrire une fonction `void supprimer_k_lettres(char s[], int k)` qui supprime les `k` premiers caractères de `s`. Si `k` est supérieur ou égal à la longueur de la chaîne, la chaîne sera réduite à la chaîne vide et `s` contiendra seulement le caractère `'\0'`.
3. On dispose d'un ensemble de points $P = \{P_i\}_{i=0}^{n-1}$. On suppose que les points P_i ont des coordonnées réelles (x_i, y_i) représentées par deux tableaux de flottants, `X` pour les abscisses, et `Y` pour les ordonnées.

- (a) Écrire une fonction qui calcule la distance entre deux points $P_i(x_i, y_i)$ et $P_j(x_j, y_j)$ ¹ selon le prototype donné :

```
double distance(double X[], double Y[], int n, int i, int j);
```

¹Soient les points $P_i(x_i, y_i)$ et $P_j(x_j, y_j)$. La distance entre P_i et P_j est donnée par l'expression $d(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. La fonction racine carrée est codée par la fonction `sqrt` de prototype `double sqrt(double x)`; (déclarée dans le fichier d'en-tête `math.h`).

- (b) Écrire une fonction qui calcule la distance minimale entre un point P_i donné et le reste des points de l'ensemble :
- ```
double distance_min_point(double X[], double Y[], int n, int i);
```
- (c) Écrire une fonction qui calcule la distance minimale entre deux points différents de l'ensemble :
- ```
double distance_min(double X[], double Y[], int n);
```
- (d) Écrire un programme `dist_min_points` qui utilise les arguments de la fonction principale `main` pour construire les tableaux des coordonnées des points. Le programme affiche les points et la distance minimale comme illustré sur l'exemple ci-dessous. Pour convertir en `double` les chaînes de caractères des arguments on utilisera la fonction `atof` (déclarée dans `stdlib.h`) de prototype : `double atof (const char *nptr);`

```
./dist-min-points
```

```
Usage: ./dist-min-points abcisse1 ordonnee1 abcisse2 ordonnee2 [...]
```

```
./dist-min-points 0. 0. 1. 0. 2. 0.
```

```
P_0=( 0.00, 0.00 )
```

```
P_1=( 1.00, 0.00 )
```

```
P_2=( 2.00, 0.00 )
```

```
La distance minimale entre une paire de points arbitraires est = 1.00
```