

Resit Exam

Duration : 1h30

Exercise 1 : (6 points)

- Write an algorithm/C program that:

- 1- Asks the user if he wants to calculate the area of a circle or a rectangle.
- 2- According to the choice of the user, the program allows the user to input the corresponding data, including the width and length for a rectangle, or the radius for a circle.
- 3- Displays the result.

- Provide a solution using a flowchart.

Exercise 2: (8 points) Write the algorithm/C program that

- a. Given a matrix of numbers ($N \times M$), it checks whether it is or not a sparse matrix (For the purpose of this exercise, we define a sparse matrix as a matrix where more than 50% of its elements are zero).
- b. When the matrix is a sparse matrix, the algorithm computes the average of the even numbers of the matrix.

Example of a sparse matrix

0	0	12	1
4	0	0	0
0	8	15	0

The average is : 8

Exercise 3 : (6 points)

Given two arrays of numbers A1 and A2 sorted into the same order, write an algorithm/C program that fuses them into a third array A sorted in the same order.

Example:

A1 : {3, 5, 8, 9, 15}

A2 : {3, 4, 10, 11, 34}

A : {3, 3, 4, 5, 8, 9, 10, 11, 15, 34}

Exercise 1 :

- Écrire un algorithme/programme C qui :

- 1- Demande à l'utilisateur s'il souhaite calculer l'aire d'un cercle ou d'un rectangle.
- 2- Selon le choix de l'utilisateur, le programme permet à l'utilisateur de saisir des données, notamment la largeur et la longueur d'un rectangle, ou le rayon d'un cercle.
- 3- Affiche le résultat.

- Propose une solution sous forme d'organigramme.

Exercise 2 : Écrire l'algorithme/C program qui

a. Étant donnée une matrice A de nombres ($N \times M$), l'algorithme vérifie s'il s'agit ou non d'une matrice creuse (Pour les besoins de cet exercice, nous définissons une matrice creuse comme une matrice dont plus de 50 % de ses éléments sont nuls).

b. Lorsque la matrice est une matrice creuse, l'algorithme/Programme C calcule la moyenne des nombres pairs de la matrice.

Exercise 3 :

Étant donné deux tableaux de nombres A1 et A2 triés dans le même ordre, écrire un algorithme/programme C qui les fusionne dans un troisième tableau A trié dans le même ordre.

Exercise n°1 :

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

```
int main() {
    int choice;
    printf("Choose the shape to calculate the area:\n");
    printf("1. Circle\n");
    printf("2. Rectangle\n");
    printf("Enter your choice (1 or 2): ");
    scanf("%d", &choice);

    if (choice == 1) {
        float radius, area;
        const PI=3.14;
        printf("Enter the radius of the circle: ");
        scanf("%f", &radius);
        area = PI * radius * radius;
        printf("The area of the circle with radius %.2f is: %.2f\n", radius, area);
    } else if (choice == 2) {
        double width, length, area;
        printf("Enter the width of the rectangle: ");
        scanf("%f", &width);
        printf("Enter the length of the rectangle: ");
        scanf("%f", &length);
        area = width * length;
        printf("The area of the rectangle with width %.2f and length %.2f is: %.2f\n", width, length, area);
    } else {
        printf("Invalid choice. Please enter 1 for circle or 2 for rectangle.\n");
    }
    return 0;
}
```

Exercise 2 :

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

```
int main() {
    int N=2, M=3;

    int matrix[N][M];
    int zeroCount = 0, evenSum = 0, evenCount = 0;
    int totalElements = N * M;
```

```
// Input matrix
printf("Enter the elements of the matrix (%d x %d):\n", N, M);
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        scanf("%d", &matrix[i][j]);
        if (matrix[i][j] == 0) {
            zeroCount++;
        }
        if (matrix[i][j] % 2 == 0) {
            evenSum += matrix[i][j];
            evenCount++;
        }
    }
}

// Check if it's a sparse matrix
if (zeroCount > totalElements / 2) {
    printf("The matrix is a sparse matrix.\n");
    if (evenCount > 0) {
        float average = (float)evenSum / evenCount;
        printf("Average of even numbers: %.2f\n", average);
    } else {
        printf("There are no even numbers in the matrix.\n");
    }
} else {
    printf("The matrix is NOT a sparse matrix.\n");
}

return 0;
}
```

Exercise 3:

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

```
int main() {
    int N1=5, N2=5;
    int N=N1+N2;

    int A1[N1], A2[N2], A[N];

    // Get elements of A1 from the user
    printf("Enter elements of A1: ");
    for (int i = 0; i < N1; i++) {
        scanf("%d", &A1[i]);
    }

    printf("Enter elements of A2: ");
    for (int i = 0; i < N2; i++) {
        scanf("%d", &A2[i]);
    }
}
```

```
}

// fusion
int i = 0, j = 0, k = 0;

while ((i < N1 || j < N2) && k < N) {
    if (A1[i] <= A2[j]) {
        A[k] = A1[i];
        i++;
    } else {
        A[k] = A2[j];
        j++;
    }
    k++;
}

// Display the fused array    optionel
printf("The fused array (A) is: ");
for (int i = 0; i < N; i++) {
    printf("%d ", A[i]);
}

return 0;
}
```