

Regular session exam

EXERCISE 01 (03 points)

Write a C program that contains a function called **findMAX()** which finds the maximum between two floating-point numbers.

Requirements:

- Define the function **findMAX()** that takes two float parameters and returns the maximum value.
- In **main()**, declare two float variables and read their values from the keyboard using **scanf()**.
- Call **findMAX()** and print the result with two decimal places.

EXERCISE 02 (06 points)

Study the following C program carefully and answer the questions below.

```
#include <stdio.h>
int computeSum(int arr[], int S);
int main() {
    int S;
    printf("Enter array size: ");
    scanf("%d", &S);
    int arr[S];
    for (int i = 0; i < S; i++) {
        printf("arr[%d] = ", i);
        scanf("%d", &arr[i]);}
    int sum = computeSum(arr, S);
    float avg = (float) sum / S;
    printf("\nSum = %d\nAverage = %.2f\n", sum, avg);
    return 0; }
int computeSum(int arr[], int S) {
    int i, sum = 0;
    for (i = 0; i < S; i++)
        sum += arr[i];
    return sum; }
```

Q1. What does this program do? Explain the role of the **computeSum()** function.

Q2. What is the output of this program if the user enters the following values:

S = 4

arr[] = {12, 8, 20, 4}

Q3. What is the purpose of the function prototype `int computeSum(int arr[], int S);` declared at the top of the program? What happens if you remove it?

EXERCISE 03 (06 points)

Write a C program to split a 1D array into two separate arrays based on positive and negative values.

Steps:

- Read the size of the original array and store it in the variable n.
- Declare three arrays: arr (original), posArr (for positive numbers), and negArr (for negative numbers).
- Read the elements of the original array arr from the user.
- If arr[i] is positive (arr[i] > 0), store it in posArr. Otherwise, store it in negArr.
- Display both posArr and negArr.

Note: You may assume that the array contains no zeros.

EXERCISE 04 (05 points)

The following C program is intended to multiply two matrices (matrix A of size 3×2 and matrix B of size 2×2), and store the result in matrix C . However, some parts of the code are missing.

```
#include <stdio.h>
int main() {
    int A[3][2] = {{10, 20}, {30,40}, {50,60}};
    int B[2][2] = {{1, 0}, {0,1}};
    int C[3][2],i, j, k;

    // Matrix multiplication: C = A * B
    for (i = 0; i < 3; i++) {
        for (_____) {
            C[i][j] = _____ ;
            for (_____) {
                C[i][j] = _____ ;
            }
        }
    }

    // Display the result
    printf("Result matrix C:\n");
    for (_____) {
        for (_____) {
            printf(______);
        }
        printf(______);
    }
    return 0;}
```

Q1. Complete the blanks (_____) to make the program work correctly.

Q2. Give the output.