Rajalakshmi Engineering College

Name: Roshan A
Email: 240701438@rajalakshmi.edu.in
Roll no: 240701438
Phone: 8754497371
Branch: REC
Department: I CSE FE
Batch: 2028
Degree: B.E - CSE

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 0_Arrays and Functions

Attempt : 1
Total Mark : 5
Marks Obtained : 5

Section 1 : Coding

1. Problem Statement

Write a program that will read a Matrix (two-dimensional arrays) and print the sum of all elements of each row by passing the matrix to a function.

Function Signature: void calculateRowSum(int [ ][ ], int, int)

Input Format

The first line consists of an integer M representing the number of rows.

The second line consists of an integer N representing the number of columns.

The next M lines consist of N space-separated integers in each line representing the elements of the matrix.

Output Format

The output displays the sum of all elements of each row separated by a space.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
3
1 2 3
4 5 6
7 8 9
Output: 6 15 24

Answer

```c
#include <stdio.h>

// You are using GCC
void calculateRowSum(int matrix[20][20], int rows, int cols) {
    //Type your code here
    for(int i=0;i<rows;i++){
        int sum=0;
        for(int j=0;j<cols;j++){
            sum+=matrix[i][j];
        }
        printf("%d ",sum);
    }
    printf("\n");
}

int main() {
    int matrix[20][20];
    int r, c;

    scanf("%d", &r);
    scanf("%d", &c);

    for (int i = 0; i < r; i++) {
        for (int j = 0; j < c; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }
```

```
   calculateRowSum(matrix, r, c);
   return 0;
}
```

Status : Correct   *Marks : 1/1*

2. Problem Statement

Alex, a budding programmer, is tasked with writing a menu-driven program
to perform operations on an array of integers. The operations include
finding the smallest number, the largest number, the sum of all numbers,
and their average. The program must repeatedly display the menu until
Alex chooses to exit.

Write a program to ensure the specified tasks are implemented based on
Alex's choices.

Input Format

The first line contains an integer n, representing the number of elements in the
array.

The second line contains n space-separated integers representing the array
elements.

The subsequent lines contain integers representing the menu choices:

Choice 1: Find and display the smallest number.

Choice 2: Find and display the largest number.

Choice 3: Calculate and display the sum of all numbers.

Choice 4: Calculate and display the average of all numbers as double.

Choice 5: Exit the program.

Output Format

For each valid menu choice, print the corresponding result:

For choice 1, print "The smallest number is: X", where X is the smallest number in the array.

For choice 2, print "The largest number is: X", where X is the largest number in the array.

For choice 3, print "The sum of the numbers is: X", where X is the sum of all numbers in the array.

For choice 4, print "The average of the numbers is: X. XX", where X.XX is the double value representing an average of all numbers in the array, rounded to two decimal places.

For choice 5, print "Exiting the program".

If an invalid choice is made, print "Invalid choice! Please enter a valid option (1-5)."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
10 20 30
1
5
Output: The smallest number is: 10
Exiting the program

Answer

```c
// You are using GCC
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int choice;
```

```c
scanf("%d",&choice);
if(choice>5){
    printf("Invalid choice! Please enter a valid option (1-5).\n");

}
while(choice!=5){
switch(choice){
    case 1:{
        int min=a[0];
        for(int i=1;i<n;i++){
            if(a[i]<min){
                min=a[i];
            }
        }
        printf(" The smallest number is: %d\n",min);
        break;}
    case 2:{
        int max=a[0];
        for(int i=1;i<n;i++){
            if(a[i]>max){
                max=a[i];
            }
        }
        printf("The largest number is: %d\n",max);
        break;}
    case 3:{
        int sum=0;
        for(int i=0;i<n;i++){
            sum+=a[i];
        }
        printf("The sum of the numbers is: %d\n",sum);
        break;}
    case 4:{
        double sum=0;
        for(int i=0;i<n;i++){
            sum+=a[i];
        }
        double avg=sum/n;
        printf("The average of the numbers is: %.2lf\n",avg);
        break;}
}
```

```
    scanf("%d",&choice);
    while(choice==5){
        printf("Exiting the program");
        break;
    }
    }

    return 0;

}
```

Status : Correct   *Marks : 1/1*

3.   Problem Statement

Tim is creating a program to track and analyze student attendance. The program requires two inputs: the total number of students (n) and the total number of class sessions (m). The task is to design and populate an attendance matrix, 'matrix', representing the attendance record of each student for each session.

The program's specific objective is to determine whether the last student on the list attended an even or odd number of classes. This functionality will aid teachers in quickly evaluating the attendance habits of individual students.

Input Format

The first line of input consists of a positive integer n, representing the number of students.

The second line consists of a positive integer m, representing the number of class sessions.

The next n lines consist of m space-separated positive integers representing the number of classes attended by the student.

Output Format

The output displays one of the following results:

If the last session is even the output prints "[LastSession] is even".

If the last session is odd the output prints "[LastSession] is odd".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 2
2
1 2
3 100
Output: 100 is even

Answer

```c
// You are using GCC
#include<stdio.h>
int main(){
    int n,m;
    scanf("%d",&n);
    scanf("%d",&m);
    int matrix[n][m];
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            scanf("%d",&matrix[i][j]);
        }
    }
    int lastsession=matrix[n - 1][m - 1];
    if(lastsession%2==0){
        printf("%d is even",lastsession);
    }
    else{
        printf("%d is odd",lastsession);
    }
}
```

Status : Correct   *Marks : 1/1*

4. Problem Statement

Saurabh is the manager of a growing tech company. He needs a program to record and analyze the monthly salaries of his employees. The program will take the number of employees and their respective salaries as input and then calculate the average salary, and find the highest and lowest salary among them.

Help Saurabh automate this task efficiently.

Input Format

The first line of input consists of an integer n, representing the number of employees.

The second line consists of n integers, where each integer represents the salary of an employee.

Output Format

The output prints n lines, where each line will display: "Employee i: "Salary

Where i is the employee number (starting from 1) and salary is the respective salary of that employee.

After that, print the average salary in the following format: "Average Salary: "average_salary

Where average_salary is the average salary of all employees, rounded to two decimal places.

Next, print the highest salary in the following format: "Highest Salary: "max_salary

Where max_salary is the highest salary among all employees.

Finally, print the lowest salary in the following format:"Lowest Salary: "min_salary

Where min_salary is the lowest salary among all employees.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
4000
3500
6000
2500
4500

Output: Employee 1: 4000
Employee 2: 3500
Employee 3: 6000
Employee 4: 2500
Employee 5: 4500

Average Salary: 4100.00
Highest Salary: 6000
Lowest Salary: 2500

Answer

```c
// You are using GCC
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++){
        printf("Employee %d: %d\n",i+1,a[i]);
    }
    double sum=0;
    int max=a[0];
```

```
    int min=a[0];
    for(int i=0;i<n;i++){
        sum+=a[i];
        if(a[i]>max){
            max=a[i];
        }
        if(a[i]<min){
            min=a[i];
        }
    }
    double avg=sum/n;
    printf("\nAverage salary: %.2f\n",avg);
    printf("Highest salary: %d\n",max);
    printf("Lowest salary: %d\n",min);
}
```

Status : Correct   *Marks : 1/1*

5. Problem Statement

Write a program that reads an integer 'n' and a square matrix of size 'n x n' from the user. The program should then set all the elements in the lower triangular part of the matrix (including the main diagonal) to zero using a function and display the resulting matrix.

Function Signature: void setZeros(int [ ][ ], int)

Input Format

The first line consists of an integer M representing the number of rows & columns.

The next M lines consist of M space-separated integers in each line representing the elements of the matrix.

Output Format

The output displays the matrix containing M space-separated elements in M lines where the lower triangular elements are replaced with zero.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
10 20 30
40 50 60
70 80 90

Output: 0 20 30
0 0 60
0 0 0

Answer

```c
#include <stdio.h>

// You are using GCC
void setZeros(int arr[10][10], int n) {
    //Type your code here
    for(int i=0;i<n;i++){
        for(int j=0;j<=i;j++){
            arr[i][j]=0;
        }
    }

}
int main() {
    int arr1[10][10];
    int n;

    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &arr1[i][j]);
        }
    }

    setZeros(arr1, n);

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", arr1[i][j]);
```

```
        }
        printf("\n");
    }

    return 0;
}
```

Status : Correct

Marks : 1/1