Labsheet :01

Name:Basanta Raj Khatiwada

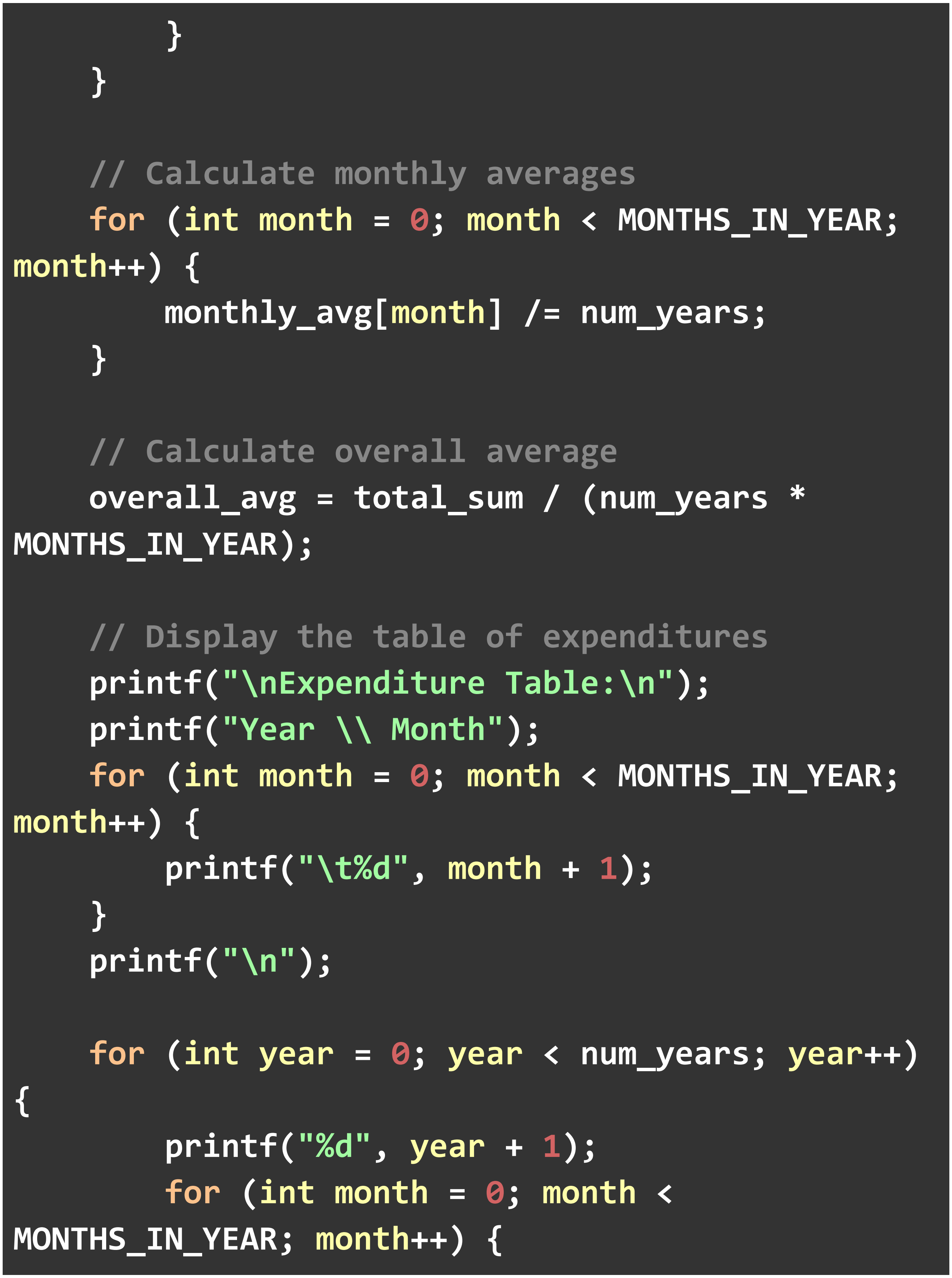
Roll No:17/bct

—---

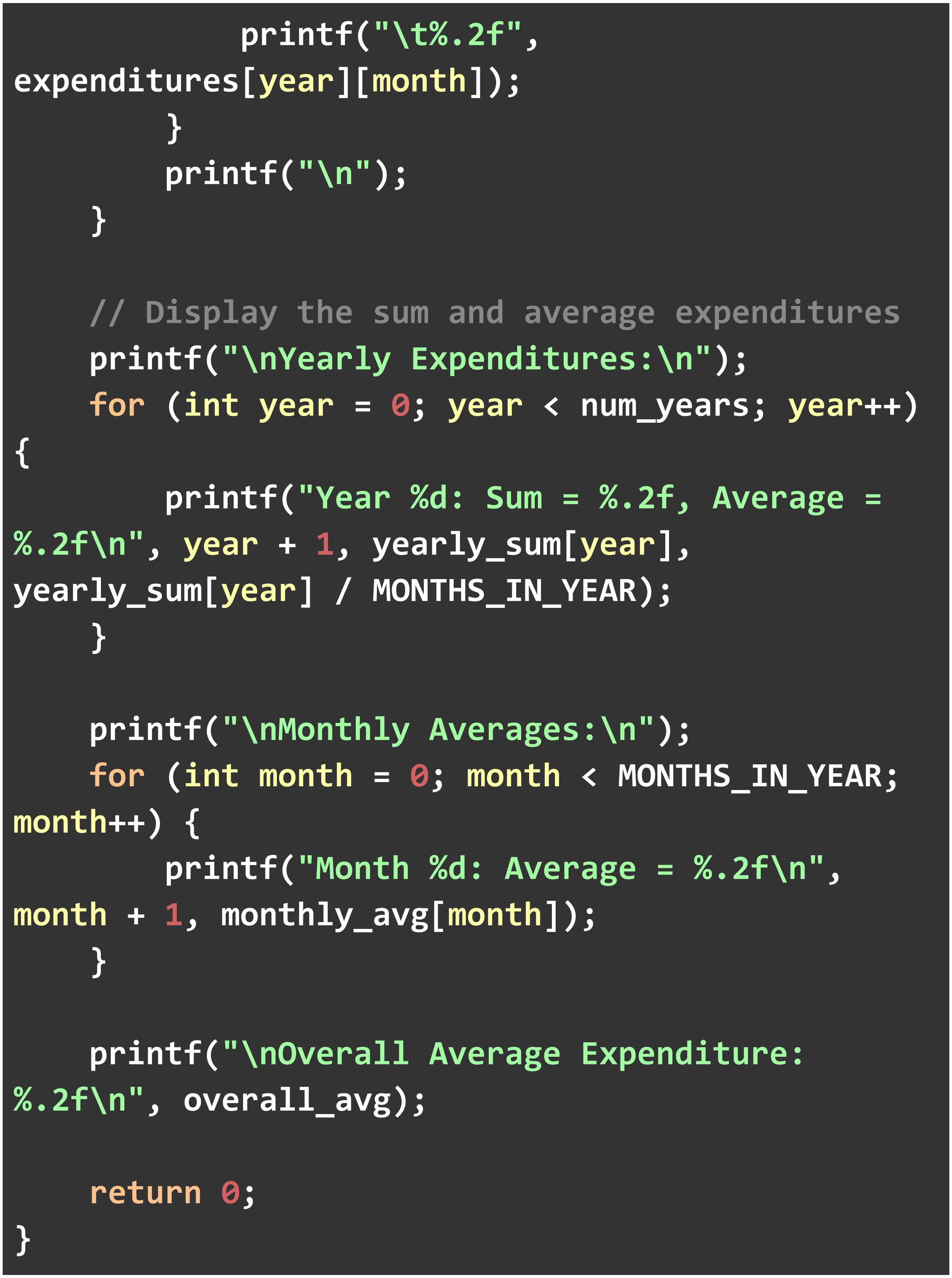
—---

1.Write a program to find the average expenditure of a company for each month of each year, each year and average over the range of years specified. Use arrays to construct a table, display the table of expenditure and find the sum and average.

```
#include <stdio.h>

#define MAX_YEARS 10
#define MONTHS_IN_YEAR 12

int main() {
    int num_years;
    float
expenditures[MAX_YEARS][MONTHS_IN_YEAR];
    float yearly_sum[MAX_YEARS] = {0};
    float monthly_avg[MONTHS_IN_YEAR] = {0};
    float total_sum = 0;
    float overall_avg;
    // Input number of years
    printf("Enter the number of years (up to
%d): ", MAX_YEARS);
    scanf("%d", &num_years);

    if (num_years > MAX_YEARS) {
        printf("Number of years exceeds the
maximum allowed (%d).\n", MAX_YEARS);
        return 1;
    }

    // Input expenditure data
    for (int year = 0; year < num_years; year++)
{
        printf("Enter the expenditures for year
%d:\n", year + 1);
        for (int month = 0; month <
MONTHS_IN_YEAR; month++) {
            printf("Month %d: ", month + 1);
            scanf("%f",
&expenditures[year][month]);
            yearly_sum[year] +=
expenditures[year][month];
            monthly_avg[month] +=
expenditures[year][month];
            total_sum +=
expenditures[year][month];
```

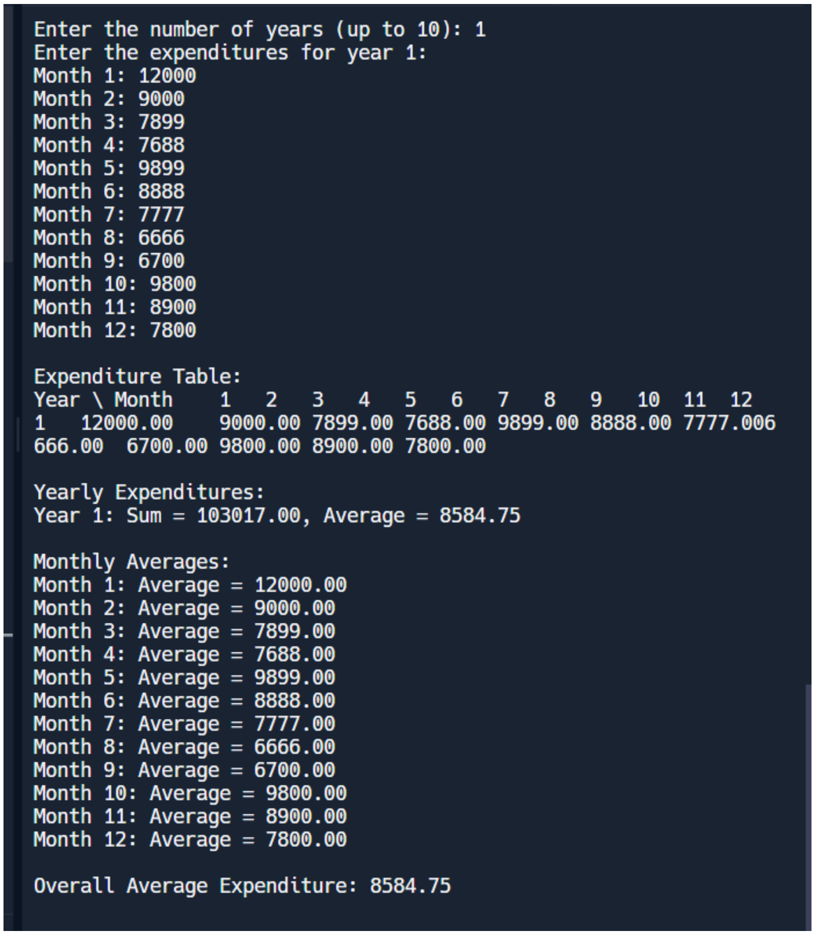


```
        }
    }

    // Calculate monthly averages
    for (int month = 0; month < MONTHS_IN_YEAR; month++) {
        monthly_avg[month] /= num_years;
    }

    // Calculate overall average
    overall_avg = total_sum / (num_years * MONTHS_IN_YEAR);

    // Display the table of expenditures
    printf("\nExpenditure Table:\n");
    printf("Year \\ Month");
    for (int month = 0; month < MONTHS_IN_YEAR; month++) {
        printf("\t%d", month + 1);
    }
    printf("\n");

    for (int year = 0; year < num_years; year++) {
        printf("%d", year + 1);
        for (int month = 0; month < MONTHS_IN_YEAR; month++) {
```



```
            printf("\t%.2f",
expenditures[year][month]);
        }
        printf("\n");
    }

    // Display the sum and average expenditures
    printf("\nYearly Expenditures:\n");
    for (int year = 0; year < num_years; year++)
{
        printf("Year %d: Sum = %.2f, Average =
%.2f\n", year + 1, yearly_sum[year],
yearly_sum[year] / MONTHS_IN_YEAR);
    }

    printf("\nMonthly Averages:\n");
    for (int month = 0; month < MONTHS_IN_YEAR;
month++) {
        printf("Month %d: Average = %.2f\n",
month + 1, monthly_avg[month]);
    }

    printf("\nOverall Average Expenditure:
%.2f\n", overall_avg);

    return 0;
}
```

Output:



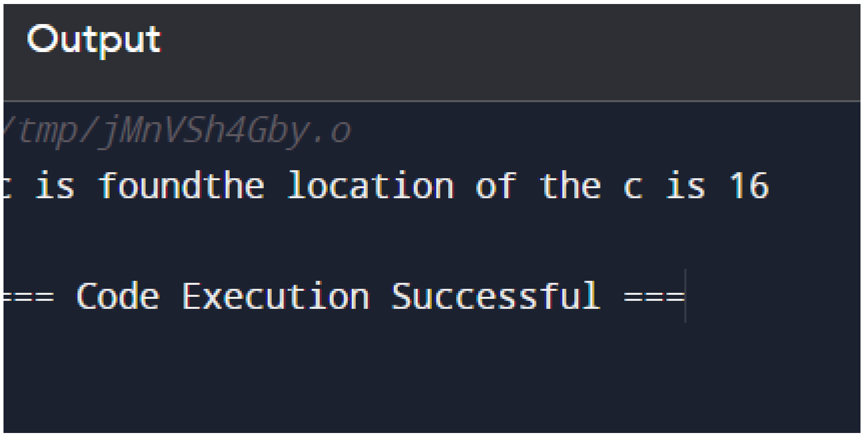
```
Enter the number of years (up to 10): 1
Enter the expenditures for year 1:
Month 1: 12000
Month 2: 9000
Month 3: 7899
Month 4: 7688
Month 5: 9899
Month 6: 8888
Month 7: 7777
Month 8: 6666
Month 9: 6700
Month 10: 9800
Month 11: 8900
Month 12: 7800

Expenditure Table:
Year \ Month    1   2   3   4   5   6   7   8   9   10  11  12
1   12000.00    9000.00 7899.00 7688.00 9899.00 8888.00 7777.006
666.00  6700.00 9800.00 8900.00 7800.00

Yearly Expenditures:
Year 1: Sum = 103017.00, Average = 8584.75

Monthly Averages:
Month 1: Average = 12000.00
Month 2: Average = 9000.00
Month 3: Average = 7899.00
Month 4: Average = 7688.00
Month 5: Average = 9899.00
Month 6: Average = 8888.00
Month 7: Average = 7777.00
Month 8: Average = 6666.00
Month 9: Average = 6700.00
Month 10: Average = 9800.00
Month 11: Average = 8900.00
Month 12: Average = 7800.00

Overall Average Expenditure: 8584.75
```

2.Write a program to find the position of the character 'C' in the sentence "idea without execution is worthless" using pointer and string.

//c code

```c
#include<stdio.h>
int main()
{
  char a[]="idea without execution is 
worthless";
  int i=0;
  while(!0){
    if(a[i]=='c'){
      printf("c is found \n");
      printf("the location of the c is %d",i+1);
      break;
    }
     i++;
  }
  return 0;
}
```

Output:

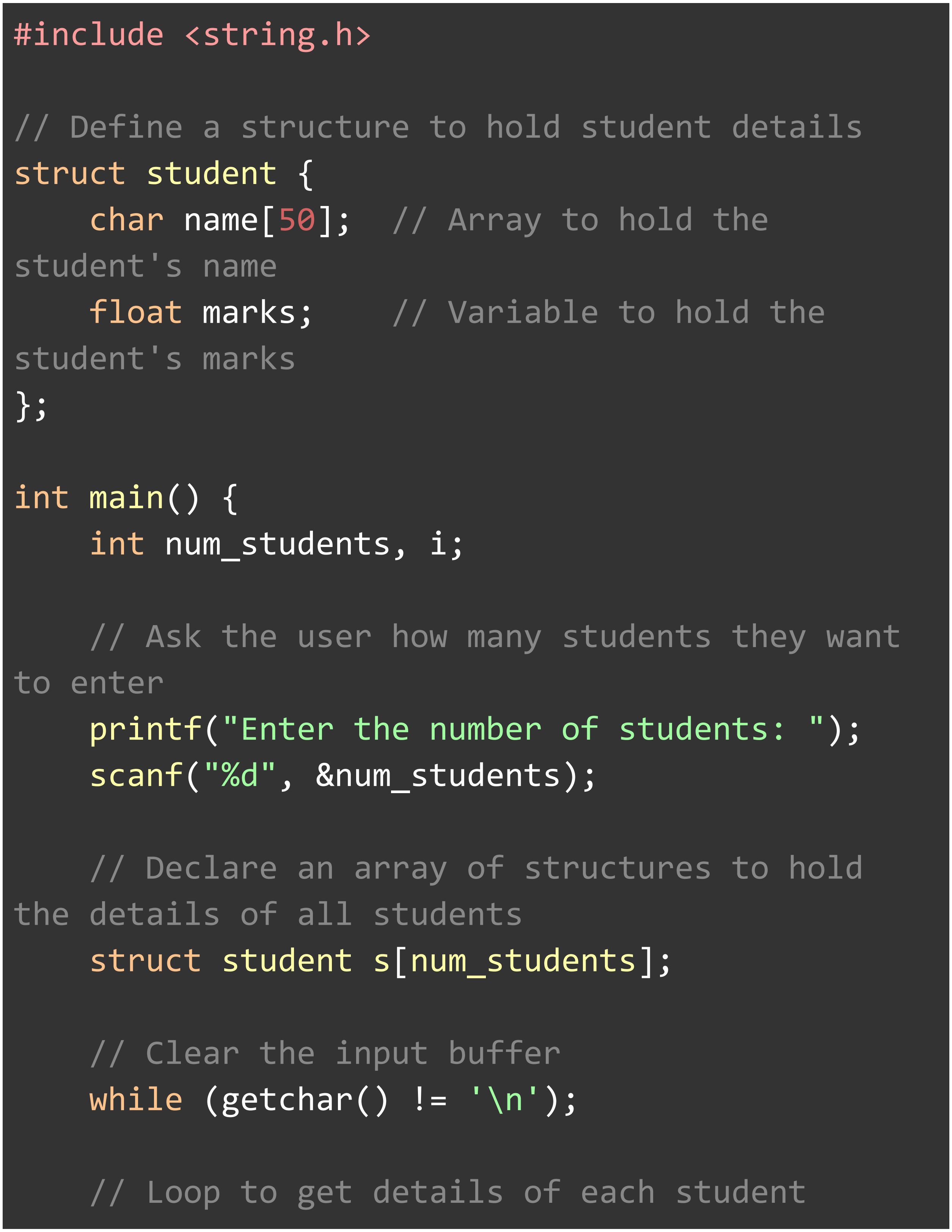


```
/tmp/jMnVSh4Gby.o
c is foundthe location of the c is 16


=== Code Execution Successful ===
```

3.Store and retrieve the name of the students and obtained marks in c programming in 1st semester using structure.

//c code

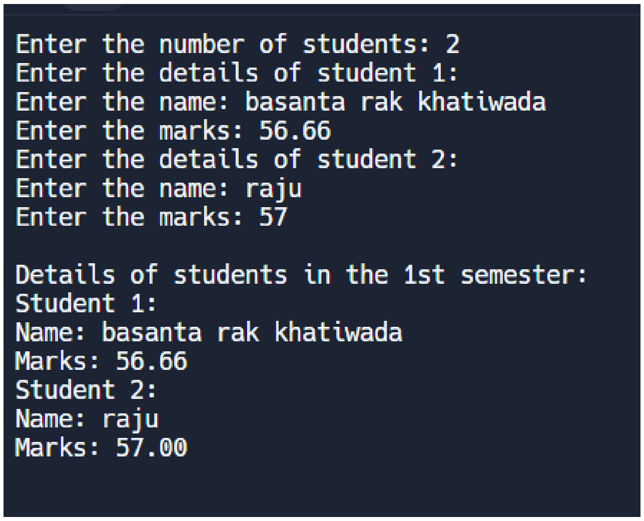
```c
#include <stdio.h>
```



```
#include <string.h>

// Define a structure to hold student details
struct student {
    char name[50];  // Array to hold the
student's name
    float marks;    // Variable to hold the
student's marks
};

int main() {
    int num_students, i;

    // Ask the user how many students they want
to enter
    printf("Enter the number of students: ");
    scanf("%d", &num_students);

    // Declare an array of structures to hold
the details of all students
    struct student s[num_students];

    // Clear the input buffer
    while (getchar() != '\n');

    // Loop to get details of each student
```

```
 for (i = 0; i < num_students; i++) {
        printf("Enter the details of student
%d:\n", i + 1);

        // Reading name with spaces
        printf("Enter the name: ");
        fgets(s[i].name, sizeof(s[i].name),
stdin);

        // Remove trailing newline character if
present
        size_t len = strlen(s[i].name);
        if (len > 0 && s[i].name[len - 1] ==
'\n') {
            s[i].name[len - 1] = '\0';
        }
        printf("Enter the marks: ");
        scanf("%f", &s[i].marks);
        while (getchar() != '\n');
    }
    printf("\nDetails of students in the 1st
semester:\n");
    for (i = 0; i < num_students; i++) {
        printf("Student %d:\n", i + 1);
        printf("Name: %s\n", s[i].name);
        printf("Marks: %.2f\n", s[i].marks);
    }
    return 0;}
```

output:



```
Enter the number of students: 2
Enter the details of student 1:
Enter the name: basanta rak khatiwada
Enter the marks: 56.66
Enter the details of student 2:
Enter the name: raju
Enter the marks: 57

Details of students in the 1st semester:
Student 1:
Name: basanta rak khatiwada
Marks: 56.66
Student 2:
Name: raju
Marks: 57.00
```

4.Write a program to read name, rollno, address, and phone number of each student in your class using structure. Store the information in file so that you can recover the information later. While recovering the information from the file sort the information alphabetically according to the name.

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

#define MAX_STUDENTS 100
#define NAME_LEN 50
#define ADDR_LEN 100
#define PHONE_LEN 15
```

```
// Define the structure to hold student
information
struct student {
    char name[NAME_LEN];
    int rollno;
    char address[ADDR_LEN];
    char phone[PHONE_LEN];
};

// Function to compare two students by name for
qsort
int compareByName(const void *a, const void *b)
{
    struct student *studentA = (struct student
*)a;
    struct student *studentB = (struct student
*)b;
    return strcmp(studentA->name,
studentB->name);
}

int main() {
    struct student students[MAX_STUDENTS];
    int num_students;
    FILE *file;

    // Input number of students
    printf("Enter the number of students: ");
    scanf("%d", &num_students);

    // Clear the input buffer
    while (getchar() != '\n');

    // Input student details
    for (int i = 0; i < num_students; i++) {
        printf("Enter details for student
%d:\n", i + 1);

        printf("Name: ");
        fgets(students[i].name, NAME_LEN,
stdin);
        // Remove trailing newline character if
present
        size_t len = strlen(students[i].name);
        if (len > 0 && students[i].name[len - 1]
== '\n') {
            students[i].name[len - 1] = '\0';
        }

        printf("Roll number: ");
        scanf("%d", &students[i].rollno);

        printf("Address: ");
        while (getchar() != '\n');  // Clear the
input buffer
        fgets(students[i].address, ADDR_LEN,
stdin);
        len = strlen(students[i].address);
        if (len > 0 && students[i].address[len -
1] == '\n') {
            students[i].address[len - 1] = '\0';
        }

        printf("Phone number: ");
        fgets(students[i].phone, PHONE_LEN,
stdin);
        len = strlen(students[i].phone);
        if (len > 0 && students[i].phone[len -
1] == '\n') {
            students[i].phone[len - 1] = '\0';
        }
    }

    // Write student details to a file
    file = fopen("students.dat", "wb");
    if (file == NULL) {
        printf("Error opening file!\n");
        return 1;
    }
    fwrite(students, sizeof(struct student),
num_students, file);
```

```
    fclose(file);

    // Recover student details from the file
    file = fopen("students.dat", "rb");
    if (file == NULL) {
        printf("Error opening file!\n");
        return 1;
    }
    fread(students, sizeof(struct student),
num_students, file);
    fclose(file);

    // Sort students by name
    qsort(students, num_students, sizeof(struct
student), compareByName);

    // Display the sorted student details
    printf("\nSorted Student Details:\n");
    for (int i = 0; i < num_students; i++) {
        printf("Name: %s\n", students[i].name);
        printf("Roll number: %d\n",
students[i].rollno);
        printf("Address: %s\n",
students[i].address);
        printf("Phone number: %s\n",
students[i].phone);
        printf("\n");
    }

    return 0;
}
```