NITHIN S
221IT085

CS111 Lab Assignment 8

Structures

Q1) C program to store Employee records (Employee ID, Name, Age, Gender, Salary) as Structures and Sort them by Salary or Employee ID

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

struct Employee {
    int empId;
    char name[50];
    int age;
    char gender;
    float salary;
};

void swap(struct Employee *a, struct Employee *b) {
    struct Employee temp = *a;
    *a = *b;
    *b = temp;
}

void bubbleSort(struct Employee *arr, int n, int
(*compare)(const void *, const void *)) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (compare(&arr[j], &arr[j + 1]) > 0) {
                swap(&arr[j], &arr[j + 1]);
            }
```

```
        }
    }
}

int compareBySalary(const void *a, const void *b) {
    return ((struct Employee *)b)->salary - ((struct
Employee *)a)->salary;
}

int compareById(const void *a, const void *b) {
    return ((struct Employee *)a)->empId - ((struct
Employee *)b)->empId;
}

int main() {
    int numEmployees;

    printf("Enter the number of employees: ");
    scanf("%d", &numEmployees);

    struct Employee *employees = (struct Employee
*)malloc(numEmployees * sizeof(struct Employee));

    for (int i = 0; i < numEmployees; ++i) {
        printf("\nEnter details for employee %d:\n",
i + 1);
        employees[i].empId = i + 1;

        printf("Name: ");
        scanf("%s", employees[i].name);

        printf("Age: ");
        scanf("%d", &employees[i].age);

        printf("Gender (M/F): ");
        scanf(" %c", &employees[i].gender);

        printf("Salary: ");
        scanf("%f", &employees[i].salary);
    }
```

```
    printf("\nUnsorted List of Employees:\n");
    printf("ID\tName\tAge\tGender\tSalary\n");
    for (int i = 0; i < numEmployees; ++i) {
        printf("%d\t%s\t%d\t%c\t%.2f\n",
employees[i].empId, employees[i].name,
employees[i].age, employees[i].gender,
employees[i].salary);
    }

    printf("\nChoose sorting option:\n");
    printf("1. Sort by Salary\n");
    printf("2. Sort by Employee ID\n");

    int choice;
    scanf("%d", &choice);

    switch (choice) {
    case 1:
        bubbleSort(employees, numEmployees,
compareBySalary);
        break;
    case 2:
        bubbleSort(employees, numEmployees,
compareById);
        break;
    default:
        printf("Invalid choice\n");
        return 1;
    }

    printf("\nSorted List of Employees:\n");
    printf("ID\tName\tAge\tGender\tSalary\n");
    for (int i = 0; i < numEmployees; ++i) {
        printf("%d\t%s\t%d\t%c\t%.2f\n",
employees[i].empId, employees[i].name,
employees[i].age, employees[i].gender,
employees[i].salary);
    }

    free(employees);
```

```
    return 0;
}
```

OUTPUT

```
nithin@nithin1729s:~/Codes/CS111/Lab_7$ gcc q1.c
nithin@nithin1729s:~/Codes/CS111/Lab_7$ ./a.out
Enter the number of employees: 3

Enter details for employee 1:
Name: Nithin
Age: 17
Gender (M/F): M
Salary: 45000

Enter details for employee 2:
Name: Harry
Age: 67
Gender (M/F): M
Salary: 7733

Enter details for employee 3:
Name: Temba
Age: 79
Gender (M/F): M
Salary: 5699

Unsorted List of Employees:
ID      Name    Age     Gender  Salary
1       Nithin  17      M       45000.00
2       Harry   67      M       7733.00
3       Temba   79      M       5699.00

Choose sorting option:
1. Sort by Salary
2. Sort by Employee ID
1

Sorted List of Employees:
ID      Name    Age     Gender  Salary
1       Nithin  17      M       45000.00
2       Harry   67      M       7733.00
3       Temba   79      M       5699.00
```

Q2) Write a C program to display student marksheet using structures. The marks sheet should print student name, reg number, 6 subjects along with marks, total marks and percentage and grade in proper format.

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


struct Student {
    char name[50];
    int regNumber;
    int marks[6];
    int totalMarks;
    float percentage;
    char grade;
};


void calculateResult(struct Student *student) {
    student->totalMarks = 0;
    for (int i = 0; i < 6; ++i) {
        student->totalMarks += student->marks[i];
    }

    student->percentage = (float)student-
>totalMarks / 6.0;

    if (student->percentage >= 90) {
        student->grade = 'A';
    } else if (student->percentage >= 80) {
        student->grade = 'B';
    } else if (student->percentage >= 70) {
        student->grade = 'C';
    } else if (student->percentage >= 60) {
        student->grade = 'D';
```

```
    } else if (student->percentage >= 50) {
        student->grade = 'E';
    } else {
        student->grade = 'F';
    }
}

int main() {

    struct Student student;


    printf("Enter Student Name: ");
    scanf("%s", student.name);

    printf("Enter Registration Number: ");
    scanf("%d", &student.regNumber);

    printf("Enter Marks for 6 Subjects:\n");
    for (int i = 0; i < 6; ++i) {
        printf("Subject %d: ", i + 1);
        scanf("%d", &student.marks[i]);
    }


    calculateResult(&student);


    printf("\nStudent Mark Sheet:\n");
    printf("Name: %s\n", student.name);
    printf("Registration Number: %d\n",
student.regNumber);
    printf("Marks in 6 Subjects:\n");
    for (int i = 0; i < 6; ++i) {
        printf("Subject %d: %d\n", i + 1,
student.marks[i]);
    }
    printf("Total Marks: %d\n", student.totalMarks);
    printf("Percentage: %.2f%%\n",
student.percentage);
    printf("Grade: %c\n", student.grade);
```

```
    return 0;
}
```

OUTPUT

```
nithin@nithin1729s:~/Codes/CS111/Lab_7$ gcc q2.c
nithin@nithin1729s:~/Codes/CS111/Lab_7$ ./a.out
Enter Student Name: Nithin
Enter Registration Number: 2210992
Enter Marks for 6 Subjects:
Subject 1: 34
Subject 2: 56
Subject 3: 76
Subject 4: 23
Subject 5: 09
Subject 6: 67

Student Mark Sheet:
Name: Nithin
Registration Number: 2210992
Marks in 6 Subjects:
Subject 1: 34
Subject 2: 56
Subject 3: 76
Subject 4: 23
Subject 5: 9
Subject 6: 67
Total Marks: 265
Percentage: 44.17%
Grade: F
nithin@nithin1729s:~/Codes/CS111/Lab_7$
```

Q3) Write a c program to generate electricity bills using structures. The bill should print customer id, First name, last name ,address, previous unit consumed, present unit utilized and display the total amount to pay to the customer.

The charge are as follow :

Unit	Charge/unit
upto 199	@1.20
200 and above but less than 400	@1.50
400 and above but less than 600	@1.80
600 and above	@2.00

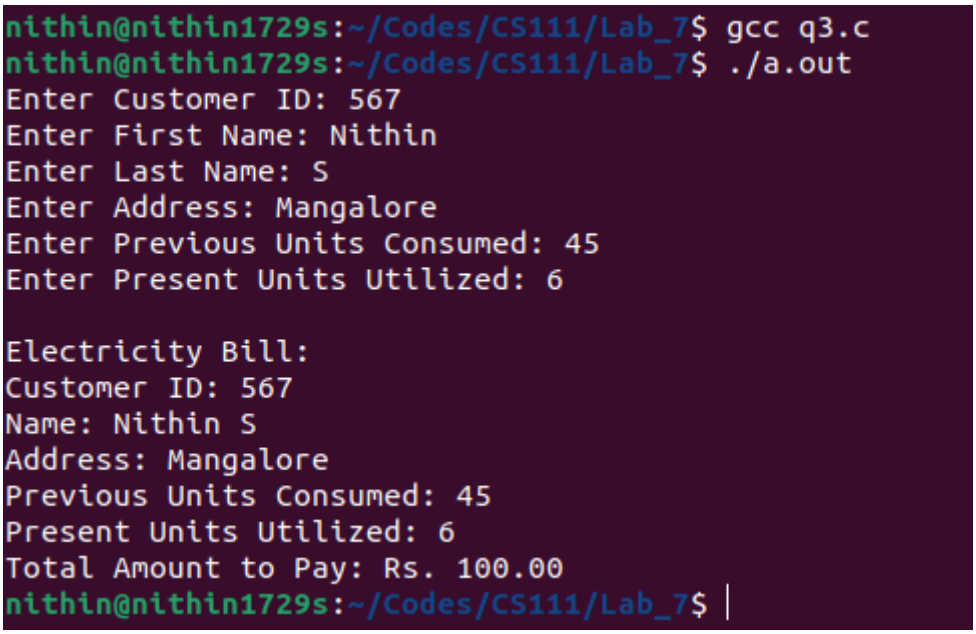
If bill exceeds Rs. 400 then a surcharge of 15% will be charged and the minimum bill should be of Rs. 100

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


struct Customer {
    int customerId;
    char firstName[50];
    char lastName[50];
    char address[100];
    int prevUnits;
    int presentUnits;
    float totalAmount;
};


void calculateBill(struct Customer *customer) {
    int unitsConsumed = customer->presentUnits -
customer->prevUnits;

    if (unitsConsumed <= 199) {
        customer->totalAmount = unitsConsumed * 1.20;
```

```
    } else if (unitsConsumed >= 200 && unitsConsumed <
400) {
        customer->totalAmount = unitsConsumed * 1.50;
    } else if (unitsConsumed >= 400 && unitsConsumed <
600) {
        customer->totalAmount = unitsConsumed * 1.80;
    } else {
        customer->totalAmount = unitsConsumed * 2.00;
    }


    if (customer->totalAmount > 400) {
        customer->totalAmount += customer->totalAmount *
0.15;
    }


    if (customer->totalAmount < 100) {
        customer->totalAmount = 100;
    }
}

int main() {

    struct Customer customer;


    printf("Enter Customer ID: ");
    scanf("%d", &customer.customerId);

    printf("Enter First Name: ");
    scanf("%s", customer.firstName);

    printf("Enter Last Name: ");
    scanf("%s", customer.lastName);

    printf("Enter Address: ");
    scanf(" %[^\n]s", customer.address);

    printf("Enter Previous Units Consumed: ");
    scanf("%d", &customer.prevUnits);

    printf("Enter Present Units Utilized: ");
    scanf("%d", &customer.presentUnits);
```

```
    calculateBill(&customer);


    printf("\nElectricity Bill:\n");
    printf("Customer ID: %d\n", customer.customerId);
    printf("Name: %s %s\n", customer.firstName,
customer.lastName);
    printf("Address: %s\n", customer.address);
    printf("Previous Units Consumed: %d\n",
customer.prevUnits);
    printf("Present Units Utilized: %d\n",
customer.presentUnits);
    printf("Total Amount to Pay: Rs. %.2f\n",
customer.totalAmount);

    return 0;
}
```

OUTPUT



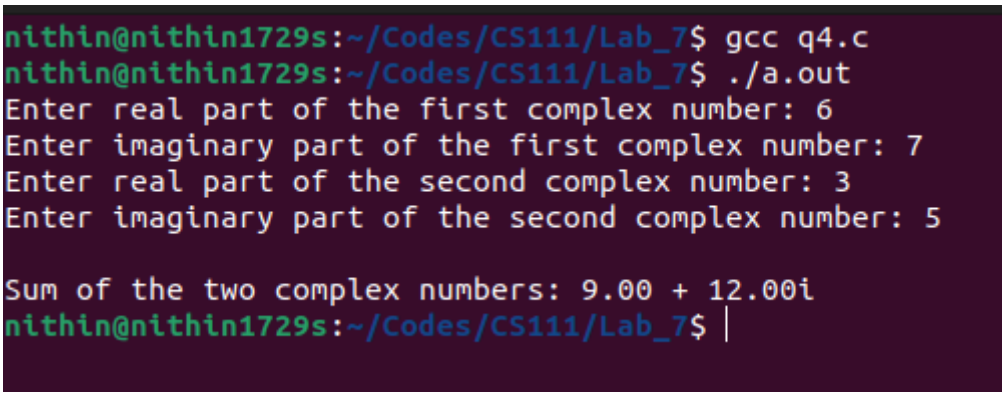
Q4) Write a program to add two complex numbers using structure.

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

```
struct Complex {
    float real;
    float imag;
};


struct Complex addComplex(struct Complex num1, struct
Complex num2) {
    struct Complex result;
    result.real = num1.real + num2.real;
    result.imag = num1.imag + num2.imag;
    return result;
}

int main() {

    struct Complex complex1, complex2, result;


    printf("Enter real part of the first complex number:
");
    scanf("%f", &complex1.real);

    printf("Enter imaginary part of the first complex
number: ");
    scanf("%f", &complex1.imag);


    printf("Enter real part of the second complex number:
");
    scanf("%f", &complex2.real);

    printf("Enter imaginary part of the second complex
number: ");
    scanf("%f", &complex2.imag);


    result = addComplex(complex1, complex2);


    printf("\nSum of the two complex numbers: %.2f +
%.2fi\n", result.real, result.imag);
```

```
    return 0;
}
```

OUTPUT



Q5) Write a C program to show usage of pointers in structure using below cases.(you may write two different programs to demonstrate each of the cases)

a) Pointer as a member of a structure

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


struct Student {
    int rollNumber;
    char *name;
    float *marks;
};


void inputStudentDetails(struct Student *student) {
    printf("Enter Roll Number: ");
    scanf("%d", &student->rollNumber);

    printf("Enter Name: ");
```

```
    student->name = (char *)malloc(50 * sizeof(char));
    scanf("%s", student->name);

    printf("Enter Marks: ");
    student->marks = (float *)malloc(sizeof(float));
    scanf("%f", student->marks);
}


void displayStudentDetails(struct Student *student) {
    printf("\nStudent Details:\n");
    printf("Roll Number: %d\n", student->rollNumber);
    printf("Name: %s\n", student->name);
    printf("Marks: %.2f\n", *student->marks);
}


void freeMemory(struct Student *student) {
    free(student->name);
    free(student->marks);
}

int main() {

    struct Student student1;


    inputStudentDetails(&student1);


    displayStudentDetails(&student1);


    freeMemory(&student1);

    return 0;
}
```

OUTPUT

```
nithin@nithin1729s:~/Codes/CS111/Lab_7$ gcc q5.c
nithin@nithin1729s:~/Codes/CS111/Lab_7$ ./a.out
Enter Roll Number: 2210992
Enter Name: Nithin
Enter Marks: 56

Student Details:
Roll Number: 2210992
Name: Nithin
Marks: 56.00
nithin@nithin1729s:~/Codes/CS111/Lab_7$
```

b) Pointer as a structure variable

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


struct Point {
    int x;
    int y;
};


void inputCoordinates(struct Point *ptr) {
    printf("Enter x-coordinate: ");
    scanf("%d", &ptr->x);

    printf("Enter y-coordinate: ");
    scanf("%d", &ptr->y);
}
```

```
void displayCoordinates(struct Point *ptr) {
    printf("\nCoordinates: (%d, %d)\n", ptr->x, ptr->y);
}

int main() {

    struct Point *pointPtr;


    pointPtr = (struct Point *)malloc(sizeof(struct
Point));


    inputCoordinates(pointPtr);


    displayCoordinates(pointPtr);


    free(pointPtr);

    return 0;
}
```

OUTPUT