\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Computer Programming Lab

CEN-392

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Program 11

Code :-

#include <stdio.h>

#include <stdbool.h>

struct Student

{

    char Name[100];

    int Roll\_No;

    float Sub\_1, Sub\_2, Sub\_3, Percentage;

};

void Insert\_Row()

{

    printf("Insert Operation Is Selected...\n");

    FILE \*fptr;

    fptr = fopen("Data.txt", "a");

    if (fptr == NULL)

    {

        printf("Error In Opening File!");

        return;

    }

    struct Student Stud;

    printf("Enter The Name : ");

    fflush(stdin);

    gets(Stud.Name);

    printf("Enter The Roll No : ");

    scanf("%d", &Stud.Roll\_No);

    printf("Enter The Marks Of Subject 1 : ");

    scanf("%f", &Stud.Sub\_1);

    printf("Enter The Marks Of Subject 2 : ");

    scanf("%f", &Stud.Sub\_2);

    printf("Enter The Marks Of Subject 3 : ");

    scanf("%f", &Stud.Sub\_3);

    Stud.Percentage = (Stud.Sub\_1 + Stud.Sub\_2 + Stud.Sub\_3) / 3;

    int num = 4;

    fwrite(&Stud, sizeof(Stud), 1, fptr);

    fclose(fptr);

    printf("\nRecord Inserted Successfully!\n");

}

void Display()

{

    printf("Display...\n");

    FILE \*fptr;

    fptr = fopen("Data.txt", "r");

    if (fptr == NULL)

    {

        printf("Error In Opening File!\n");

        return;

    }

    struct Student Temp;

    printf("| Name | Roll No | Subject 1 | Subject 2 | Subject 3 | Percentage |\n\n");

    while (fread(&Temp, sizeof(Temp), 1, fptr))

    {

        printf("%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\n", Temp.Name, Temp.Roll\_No, Temp.Sub\_1, Temp.Sub\_2, Temp.Sub\_3, Temp.Percentage);

    }

    fclose(fptr);

}

void Remove\_Row()

{

    FILE \*fptr = NULL, \*tptr = NULL;

    fptr = fopen("Data.txt", "r");

    if (fptr == NULL)

    {

        printf("Error In Opening File!\n");

        return;

    }

    tptr = fopen("temp.txt", "a");

    printf("Remove Operation Is Selected...\n");

    int Roll\_No;

    printf("Enter The Roll No Of Student : ");

    scanf("%d", &Roll\_No);

    struct Student Temp;

    bool Found = false;

    while (fread(&Temp, sizeof(Temp), 1, fptr))

    {

        if (Roll\_No == Temp.Roll\_No)

        {

            Found = true;

            continue;

        }

        fwrite(&Temp, sizeof(Temp), 1, tptr);

    }

    fclose(fptr);

    fclose(tptr);

    remove("Data.txt");

    rename("temp.txt", "Data.txt");

    if (Found == false)

        printf("\nNo Such Roll No Found In Data Base\n");

    else

        printf("\nRow Successfully Removed!\n");

}

void Update\_Row()

{

    FILE \*fptr = NULL, \*tptr = NULL;

    fptr = fopen("Data.txt", "r");

    if (fptr == NULL)

    {

        printf("Error In Opening File!\n");

        return;

    }

    tptr = fopen("temp.txt", "a");

    printf("Update Operation Is Selected...\n");

    int Roll\_No;

    printf("Enter The Roll No Of Student : ");

    scanf("%d", &Roll\_No);

    struct Student Temp;

    bool Found = false;

    while (fread(&Temp, sizeof(Temp), 1, fptr))

    {

        if (Roll\_No == Temp.Roll\_No)

        {

            Found = true;

            printf("Enter The Name : ");

            fflush(stdin);

            gets(Temp.Name);

            printf("Enter The Roll No : ");

            scanf("%d", &Temp.Roll\_No);

            printf("Enter The Marks Of Subject 1 : ");

            scanf("%f", &Temp.Sub\_1);

            printf("Enter The Marks Of Subject 2 : ");

            scanf("%f", &Temp.Sub\_2);

            printf("Enter The Marks Of Subject 3 : ");

            scanf("%f", &Temp.Sub\_3);

            Temp.Percentage = (Temp.Sub\_1 + Temp.Sub\_2 + Temp.Sub\_3) / 3;

        }

        fwrite(&Temp, sizeof(Temp), 1, tptr);

    }

    fclose(fptr);

    fclose(tptr);

    remove("Data.txt");

    rename("temp.txt", "Data.txt");

    if (Found == false)

        printf("\nNo Such Roll No Found In Data Base\n");

    else

        printf("\nRow Successfully Updated!\n");

}

void Add\_Bars()

{

    printf("-----------------------------------------------------------------------\n");

}

void Menu()

{

    printf("\_\_\_Operation\_\_\_\n");

    printf("1.Insert Row\n");

    printf("2.Remove Row\n");

    printf("3.Update Row\n");

    printf("4.Display\n");

    printf("5.Exit\n\n");

    printf("Enter Your Choice : ");

}

int Options()

{

    int opt;

    fflush(stdin);

    scanf("%d", &opt);

    Add\_Bars();

    switch (opt)

    {

    case 1:

        Insert\_Row();

        break;

    case 2:

        Remove\_Row();

        break;

    case 3:

        Update\_Row();

        break;

    case 4:

        Display();

        break;

    case 5:

        return 0;

    default:

        printf("Incorrect Input!\nTry Again!\n");

        break;

    }

    Add\_Bars();

    return 1;

}

int main()

{

    system("cls");

    printf("\_\_\_Vicky\_Gupta\_20BCS070\_\_\_\n\n");

    while (1)

    {

        Menu();

        if (!Options())

            break;

    }

    printf("Exiting...");

    Add\_Bars();

    return 0;

}

Output :-

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated