Write a C program to create a student record using structure. Take the structure members as student-name, student-roll-no, branch, aggregate-percentage and address. Create a single structure variable (of student type) and enter the students details and display it on screen.

#include<stdio.h>

struct student

{

    char student\_name[20];

    int student\_roll\_no;

    char branch[10];

    float aggregate\_percentage;

    char address[20];

};

void main()

{

    struct student s;

    printf("Enter the following details\n");

    printf("Name: ");

    scanf("%s", s.student\_name);

    printf("Roll number: ");

    scanf("%d", &s.student\_roll\_no);

    printf("Branch: ");

    scanf("%s", s.branch);

    printf("Percentage: ");

    scanf("%f", &s.aggregate\_percentage);

    printf("Address: ");

    scanf("%s", s.address);

    printf("\nThe details of the student are\n");

    printf("Name: %s\n", s.student\_name);

    printf("Roll number: %d\n", s.student\_roll\_no);

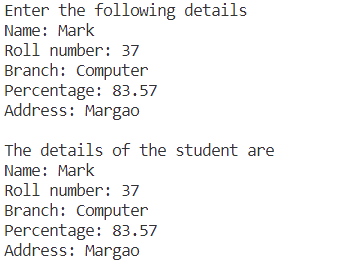
    printf("Branch: %s\n", s.branch);

    printf("Percentage: %f\n", s.aggregate\_percentage);

    printf("Address: %s\n", s.address);

}

Output



Write a C program to create an array of structure variables (for the above question). Maintain 5 students’ records. Provide following menu driven options in your program.

1. Enter student details

2. Display students details

3. Search student based on student-roll-no

4. Display student details whose aggregate-percentage is above 70

#include <stdio.h>

struct student

{

    char student\_name[20];

    int student\_roll\_no;

    char branch[10];

    float aggregate\_percentage;

    char address[20];

};

struct student s[5];

void enter()

{

    for (int i = 0; i < 5; i++)

    {

        printf("\nEnter the following details for student no. %d\n", i + 1);

        printf("Name: ");

        scanf("%s", s[i].student\_name);

        printf("Roll number: ");

        scanf("%d", &s[i].student\_roll\_no);

        printf("Branch: ");

        scanf("%s", s[i].branch);

        printf("Percentage: ");

        scanf("%f", &s[i].aggregate\_percentage);

        printf("Address: ");

        scanf("%s", s[i].address);

    }

}

void display()

{

    for (int i = 0; i < 5; i++)

    {

        printf("\nThe details of student no. %d are\n", i + 1);

        printf("Name: %s\n", s[i].student\_name);

        printf("Roll number: %d\n", s[i].student\_roll\_no);

        printf("Branch: %s\n", s[i].branch);

        printf("Percentage: %f\n", s[i].aggregate\_percentage);

        printf("Address: %s\n", s[i].address);

    }

}

void search()

{

    int roll, f = 0;

    printf("Enter the roll number: ");

    scanf("%d", &roll);

    for(int i = 0; i < 5; i++)

    {

        if(s[i].student\_roll\_no == roll)

        {

            printf("\nThe details of the student are\n");

            printf("Name: %s\n", s[i].student\_name);

            printf("Branch: %s\n", s[i].branch);

            printf("Percentage: %f\n", s[i].aggregate\_percentage);

            printf("Address: %s\n", s[i].address);

            f++;

            break;

        }

    }

    if(f == 0)

        printf("Student not found\n");

}

void display\_70\_plus()

{

    printf("The students scoring above 70 percent are\n");

    for(int i = 0; i < 5; i++)

    {

        if(s[i].aggregate\_percentage > 70)

        {

            printf("Name: %s\n", s[i].student\_name);

            printf("Roll number: %d\n", s[i].student\_roll\_no);

            printf("Branch: %s\n", s[i].branch);

            printf("Percentage: %f\n", s[i].aggregate\_percentage);

            printf("Address: %s\n\n", s[i].address);

        }

    }

}

int main()

{

    int p;

    do

    {

        printf("1. Enter student details\n");

        printf("2. Display students details\n");

        printf("3. Search student based on student-roll-no\n");

        printf("4. Display student details whose aggregate-percentage is above 70\n");

        printf("5. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &p);

        switch (p)

        {

            case 1:

                enter();

                break;

            case 2:

                display();

                break;

            case 3:

                search();

                break;

            case 4:

                display\_70\_plus();

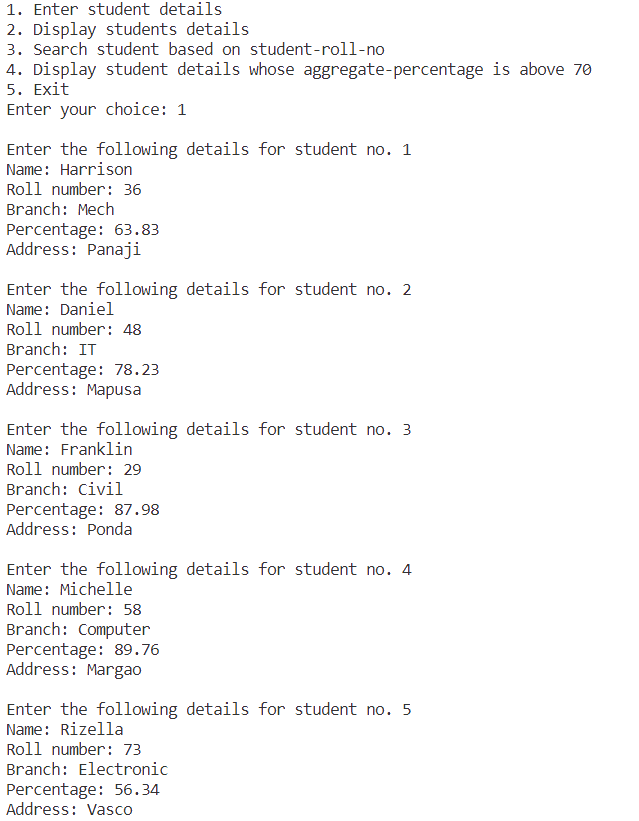
                break;

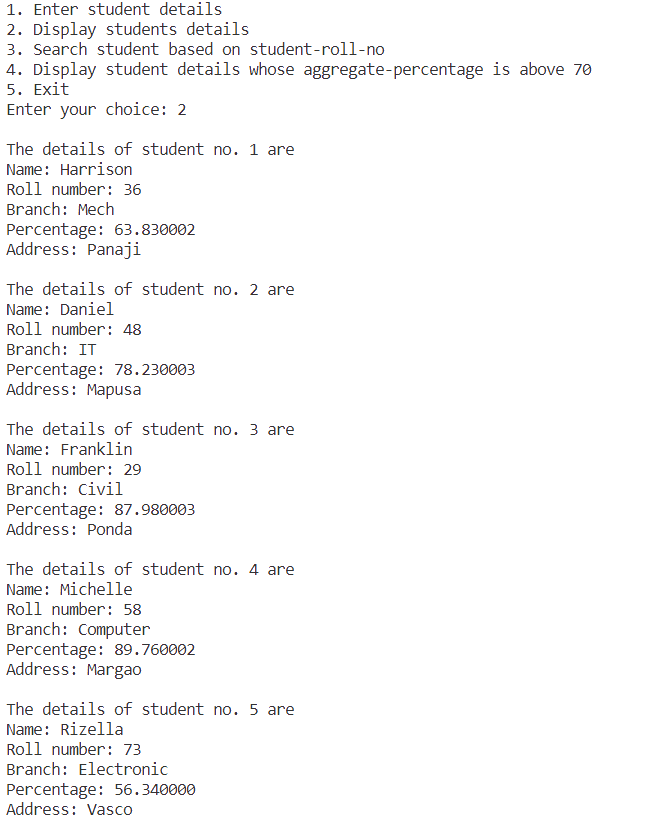
        }

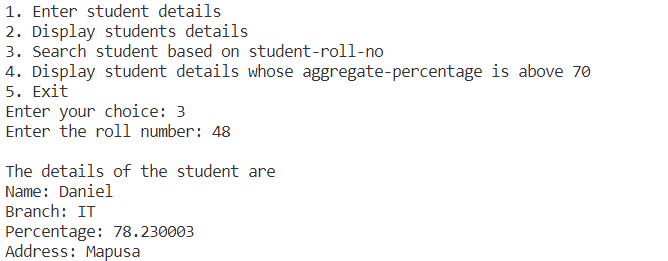
    } while (p != 5);

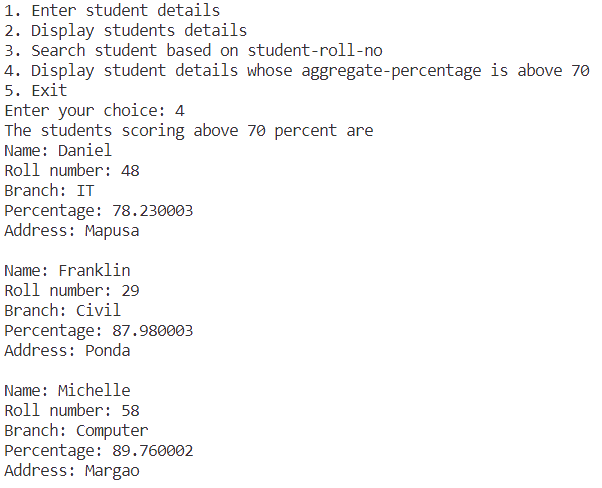
    return 0;

}









Write a C program to demonstrate how to pass a copy of structure record to an user defined function. (Take your own example)

#include<stdio.h>

struct account

{

    int account\_number;

    float balance;

};

struct account balance\_update(struct account xyz, float p)

{

    xyz.balance += p;

    return (xyz);

}

int main()

{

    struct account savings;

    float p;

    printf("Enter account number and existing balance\n");

    scanf("%d %f", &savings.account\_number, &savings.balance);

    printf("Enter amount deposited: ");

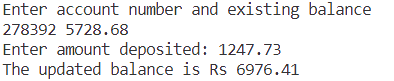
    scanf("%f", &p);

    savings = balance\_update(savings, p);

    printf("The updated balance is Rs %.2f\n", savings.balance);

}

Output



Write a C program to demonstrate the size of a union type record.(Use the sizeof operator) Take your own example.

#include<stdio.h>

union subject

{

   int marks;

   float percentage;

   char grade;

};

int main()

{

    int size = sizeof(union subject);

    printf("Size of union subject is %d", size);

}

Output

