1. Define a structure for student record and print details.

IPO

**Input:** The program takes as input the first mark (mark1), the roll number (rollno), the second mark (mark2), an initial total (ignored in calculation), an initial average (ignored in calculation), and the student’s name.

**Process:** It recalculates the total as the sum of mark1 and mark2, and then computes the average by dividing the total by 2 using float division to preserve decimals.

**Output:** It displays the entered marks, roll number, the calculated total, the calculated average, and the student’s name.

CODE

#include<stdio.h>

struct student

{

int mark1,rollno,mark2,total;

float avg;

char name[20];

}s1;

void main()

{

scanf("%d%d%d%d%f",&s1.mark1,&s1.rollno,&s1.mark2,&s1.total,&s1.avg);

scanf("%s",s1.name);

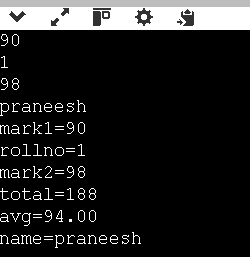
s1.total=s1.mark1+s1.mark2;

s1.avg=s1.total/2;

printf("mark1=%d\nrollno=%d\nmark2=%d\ntotal=%d\navg=%.2f\nname=%s\n",s1.mark1,s1.rollno,s1.mark2,s1.total,s1.avg,s1.name);

}

OUTPUT



1. Write a program to store and display employee details using structures.

IPO

**Input:** The program takes the employee ID, salary, years of experience, and employee name as input from the user.

**Process:** It stores the entered values in the members of a structure variable s1.

**Output:** It displays the employee ID, salary, years of experience, and employee name.

CODE

#include<stdio.h>

struct student

{

int employeeid,salary;

char employeename[20];

int experience;

}s1;

void main()

{

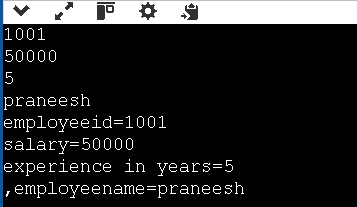
scanf("%d%d%d",&s1.employeeid,&s1.salary,&s1.experience);

scanf("%s",s1.employeename);

printf("employeeid=%d\nsalary=%d\nexperience in years=%d\n,employeename=%s\n",s1.employeeid,s1.salary,s1.experience,s1.employeename);

}

OUTPUT



1. Write a program to pass a structure to a function.

IPO  
**Input:** The program takes a student’s roll number, name, and marks as input from the user.

**Process:** It stores the entered details in a structure variable, then passes this structure to a function that displays the stored information.

**Output:** The program displays the student’s roll number, name, and marks.

Ask ChatGPT

CODE

#include <stdio.h>

struct student {

int rollno;

char name[20];

float marks;

};

void display(struct student s) {

printf("Roll No: %d\n", s.rollno);

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

printf("Marks: %.2f\n", s.marks);

}

void main() {

struct student s1;

printf("Enter roll number: ");

scanf("%d", &s1.rollno);

printf("Enter name: ");

scanf("%s", s1.name);

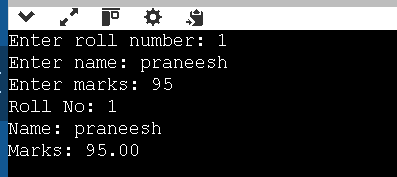
printf("Enter marks: ");

scanf("%f", &s1.marks);

display(s1);

}

OUTPUT



1. Write a program to store multiple student records using array of structures.

IPO

**Input:** The program takes the number of students, then for each student takes roll number, name, and marks as input.

**Process:** It stores each student’s details in separate elements of an array of structures, preserving all records.

**Output:** The program displays all stored student records with roll number, name, and marks.

CODE

#include <stdio.h>

struct student {

int rollno;

char name[20];

float marks;

};

void main() {

struct student s[100];

int n, i;

printf("Enter number of students: ");

scanf("%d", &n);

for (i = 0; i < n; i++) {

printf("\nEnter details of student %d:\n", i + 1);

printf("Roll Number: ");

scanf("%d", &s[i].rollno);

printf("Name: ");

scanf("%s", s[i].name);

printf("Marks: ");

scanf("%f", &s[i].marks);

}

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

for (i = 0; i < n; i++) {

printf("\nStudent %d:\n", i + 1);

printf("Roll No: %d\n", s[i].rollno);

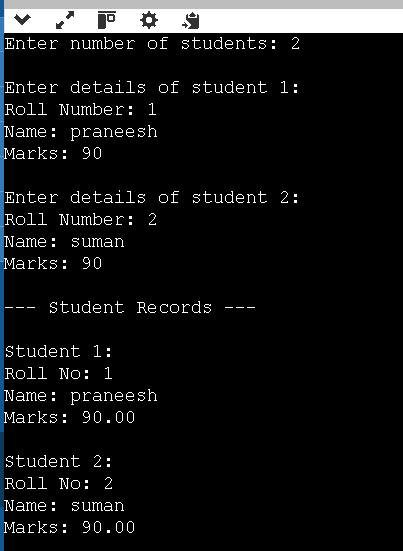
printf("Name: %s\n", s[i].name);

printf("Marks: %.2f\n", s[i].marks);

}

}

OUTPUT



1. Write a program to demonstrate nested structures.

IPO

**Input:** The program takes a student’s roll number, name, city, and pincode as input.

**Process:** It stores the student’s roll number and name in the outer student structure, and stores the city and pincode in the nested address structure inside it.

**Output:** It displays the student’s roll number, name, city, and pincode.

CODE

#include <stdio.h>

struct address {

char city[20];

int pincode;

};

struct student {

int rollno;

char name[20];

struct address addr; // Nested structure

};

void main() {

struct student s1;

printf("Enter roll number: ");

scanf("%d", &s1.rollno);

printf("Enter name: ");

scanf("%s", s1.name);

printf("Enter city: ");

scanf("%s", s1.addr.city);

printf("Enter pincode: ");

scanf("%d", &s1.addr.pincode);

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

printf("Roll No: %d\n", s1.rollno);

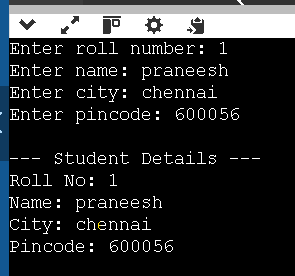
printf("Name: %s\n", s1.name);

printf("City: %s\n", s1.addr.city);

printf("Pincode: %d\n", s1.addr.pincode);

}

OUTPUT



1. Write a program to calculate total and average marks using structures.

IPO

**Input:** The program takes as input the first mark (mark1), the roll number (rollno), the second mark (mark2), an initial total (ignored in calculation), an initial average (ignored in calculation), and the student’s name.

**Process:** It recalculates the total as the sum of mark1 and mark2, and then computes the average by dividing the total by 2 using float division to preserve decimals.

**Output:** It displays the entered marks, roll number, the calculated total, the calculated average, and the student’s name.

CODE

#include<stdio.h>

struct student

{

int mark1,rollno,mark2,total;

float avg;

char name[20];

}s1;

void main()

{

scanf("%d%d%d%d%f",&s1.mark1,&s1.rollno,&s1.mark2,&s1.total,&s1.avg);

scanf("%s",s1.name);

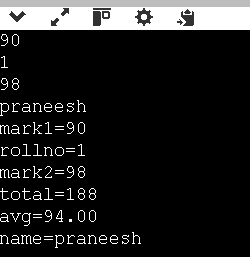
s1.total=s1.mark1+s1.mark2;

s1.avg=s1.total/2;

printf("mark1=%d\nrollno=%d\nmark2=%d\ntotal=%d\navg=%.2f\nname=%s\n",s1.mark1,s1.rollno,s1.mark2,s1.total,s1.avg,s1.name);

}

OUTPUT



1. Write a program to find the highest marks among students.

IPO

**Input:** The program takes the number of students, and for each student takes roll number, name, and marks.

**Process:** It stores all student details in an array of structures, compares their marks while reading the data, and identifies the student with the highest marks.

**Output:** It displays the roll number, name, and marks of the student who scored the highest.

CODE

#include <stdio.h>

struct student {

int rollno;

char name[20];

float marks;

};

void main() {

struct student s[100];

int n, i;

int highestIndex = 0;

printf("Enter number of students: ");

scanf("%d", &n);

for (i = 0; i < n; i++) {

printf("\nEnter details of student %d:\n", i + 1);

printf("Roll Number: ");

scanf("%d", &s[i].rollno);

printf("Name: ");

scanf("%s", s[i].name);

printf("Marks: ");

scanf("%f", &s[i].marks);

if (s[i].marks > s[highestIndex].marks) {

highestIndex = i;

}

}

printf("\n--- Student with Highest Marks ---\n");

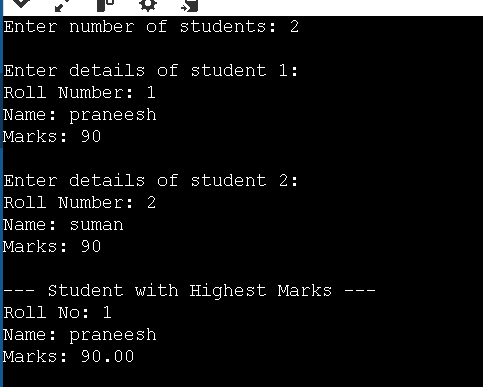
printf("Roll No: %d\n", s[highestIndex].rollno);

printf("Name: %s\n", s[highestIndex].name);

printf("Marks: %.2f\n", s[highestIndex].marks);

}

OUTPUT



1. Write a program to sort student records by name using structure.

IPO

**Input:** The program takes the number of students, and for each student, it takes roll number, name, and marks.

**Process:** It stores all student details in an array of structures, then sorts the array in ascending alphabetical order of names using string comparison (strcmp) and swapping.

**Output:** It displays all student records sorted by name.

CODE

#include <stdio.h>

#include <string.h>

struct student {

int rollno;

char name[20];

float marks;

};

void main() {

struct student s[100], temp;

int n, i, j;

printf("Enter number of students: ");

scanf("%d", &n);

for (i = 0; i < n; i++) {

printf("\nEnter details of student %d:\n", i + 1);

printf("Roll Number: ");

scanf("%d", &s[i].rollno);

printf("Name: ");

scanf("%s", s[i].name);

printf("Marks: ");

scanf("%f", &s[i].marks);

}

for (i = 0; i < n - 1; i++) {

for (j = i + 1; j < n; j++) {

if (strcmp(s[i].name, s[j].name) > 0) {

temp = s[i];

s[i] = s[j];

s[j] = temp;

}

}

}

printf("\n--- Student Records Sorted by Name ---\n");

for (i = 0; i < n; i++) {

printf("Roll No: %d\n", s[i].rollno);

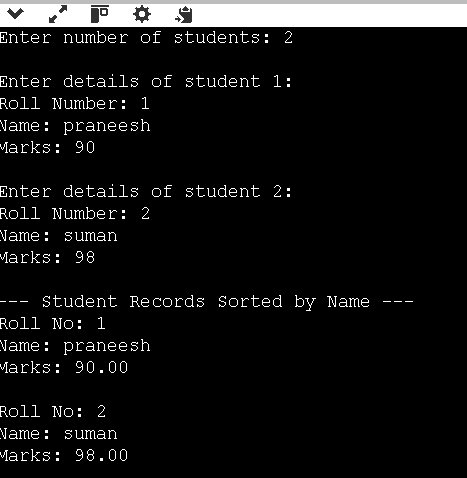
printf("Name: %s\n", s[i].name);

printf("Marks: %.2f\n\n", s[i].marks);

}

}

OUTPUT



9. Write a program using union to store data of different types.  
IPO

**Input:** The program assigns an integer, a float, and a string to the members of a union one by one.

**Process:** It stores each value in the same memory location of the union, overwriting the previous value each time.

**Output:** It displays the integer, then the float, then the string — but only the most recently stored value is valid.

CODE

#include <stdio.h>

union Data {

int i;

float f;

char str[20];

};

void main() {

union Data data;

data.i = 10;

printf("Integer: %d\n", data.i);

data.f = 3.14;

printf("Float: %.2f\n", data.f);

sprintf(data.str, "Hello");

printf("String: %s\n", data.str);

}

OUTPUT



10. Compare and contrast structure vs union with a sample program.

IPO

**Input:** The program assigns values to members of both a structure and a union.

**Process:** For the structure, all members store their own values independently. For the union, only the last assigned member retains a valid value because all members share the same memory.

**Output:** The program displays that in a structure, all values remain intact, while in a union, only the most recent value is valid.

CODE

#include <stdio.h>

#include <string.h>

struct StudentStruct {

int rollno;

float marks;

char name[20];

};

union StudentUnion {

int rollno;

float marks;

char name[20];

};

void main() {

struct StudentStruct s1;

union StudentUnion u1;

s1.rollno = 101;

s1.marks = 88.5;

strcpy(s1.name, "Rahul");

printf("Structure:\n");

printf("Roll No: %d\n", s1.rollno);

printf("Marks: %.2f\n", s1.marks);

printf("Name: %s\n", s1.name);

u1.rollno = 101;

printf("\nUnion after rollno:\nRoll No: %d\n", u1.rollno);

u1.marks = 88.5;

printf("Union after marks:\nMarks: %.2f\n", u1.marks);

strcpy(u1.name, "Rahul");

printf("Union after name:\nName: %s\n", u1.name);

}

OUTPUT

