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

**Input:**

* Student roll number, name, and marks

**🔹 Process:**

* Use a structure to store the data
* Assign values to structure members

**🔹 Output:**

* Print the stored student details

#include <stdio.h>

struct Student {

int rollNo;

char name[50];

float marks;

};

int main() {

struct Student s1;

// Input student details

printf("Enter student roll number: ");

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

printf("Enter student name: ");

scanf(" %[^\n]", s1.name);

printf("Enter student marks: ");

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

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

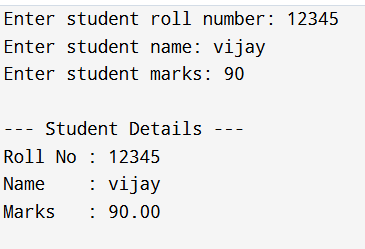
printf("Roll No : %d\n", s1.rollNo);

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

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

return 0;

}



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

**Input:**

* Employee ID
* Employee Name
* Employee Salary

**🔹 Process:**

* Define a structure to hold employee information
* Use variables to store and manage the structure data

**🔹 Output:**

* Display employee details stored in the structure

#include <stdio.h>

struct Employee {

int empID;

char name[50];

float salary;

};

int main() {

struct Employee emp;

printf("Enter Employee ID: ");

scanf("%d", &emp.empID);

printf("Enter Employee Name: ");

scanf(" %[^\n]", emp.name);

printf("Enter Employee Salary: ");

scanf("%f", &emp.salary);

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

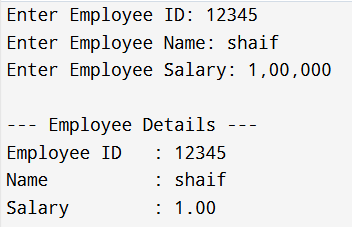
printf("Employee ID : %d\n", emp.empID);

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

printf("Salary : %.2f\n", emp.salary);

return 0;

}



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

**Input:**

* Roll number, name, and marks of a student

**🔹 Process:**

* Define a structure Student
* Read values and store them in a structure variable
* Pass the structure variable to a function to display details

**🔹 Output:**

* Print the student details using the function

#include <stdio.h>

struct Student {

int rollNo;

char name[50];

float marks;

};

void displayStudent(struct Student s) {

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

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

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

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

}

int main() {

struct Student s1;

printf("Enter Roll Number: ");

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

printf("Enter Name: ");

scanf(" %[^\n]", s1.name);

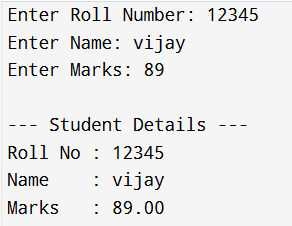
printf("Enter Marks: ");

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

displayStudent(s1);

return 0;

}



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

**Input:**

* Roll number, name, and marks of 3 students

**🔹 Process:**

* Use an array of struct Student
* Loop through array to input and store data
* Loop again to display each student’s data

**🔹 Output:**

* Print the student records for all 3 students

#include <stdio.h>

struct Student {

int rollNo;

char name[50];

float marks;

};

int main() {

struct Student students[3];

int i;

printf("Enter details for 3 students:\n");

for (i = 0; i < 3; i++) {

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

printf("Enter Roll No: ");

scanf("%d", &students[i].rollNo);

printf("Enter Name: ");

scanf(" %[^\n]", students[i].name);

printf("Enter Marks: ");

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

}

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

for (i = 0; i < 3; i++) {

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

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

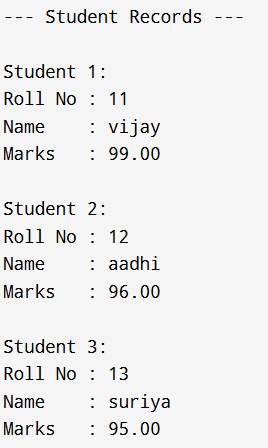
printf("Name : %s\n", students[i].name);

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

}

return 0;

}



1. Write a program to demonstrate nested structures.

**Input:**

* Roll number, name, and date of birth (day, month, year)

**🔹 Process:**

* Define struct Date for date
* Define struct Student that includes struct Date
* Store and access nested data using dot notation (s.dob.day)

**🔹 Output:**

* Display student details including nested date of birth

#include <stdio.h>

struct Date {

int day;

int month;

int year;

};

struct Student {

int rollNo;

char name[50];

struct Date dob;

};

int main() {

struct Student s;

printf("Enter roll number: ");

scanf("%d", &s.rollNo);

printf("Enter name: ");

scanf(" %[^\n]", s.name);

printf("Enter date of birth (DD MM YYYY): ");

scanf("%d %d %d", &s.dob.day, &s.dob.month, &s.dob.year);

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

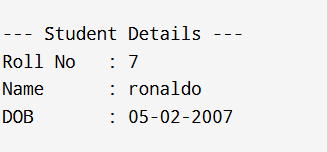
printf("Roll No : %d\n", s.rollNo);

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

printf("DOB : %02d-%02d-%04d\n", s.dob.day, s.dob.month, s.dob.year);

return 0;

}



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

**Input:**

* Student name
* Marks in 3 subjects

**🔹 Process:**

* Use a structure to store marks
* Add all marks to get total
* Calculate average as total / 3.0

**🔹 Output:**

* Display student name, total, and average marks

#include <stdio.h>

struct Student {

char name[50];

int marks[3];

int total;

float average;

};

int main() {

struct Student s;

int i;

printf("Enter student name: ");

scanf(" %[^\n]", s.name);

s.total = 0;

printf("Enter marks for 3 subjects:\n");

for (i = 0; i < 3; i++) {

printf("Subject %d: ", i + 1);

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

s.total += s.marks[i];

}

s.average = s.total / 3.0;

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

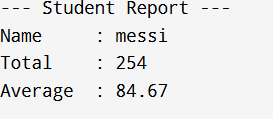
printf("Name : %s\n", s.name);

printf("Total : %d\n", s.total);

printf("Average : %.2f\n", s.average);

return 0;

}



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

**Input:**

* Number of students
* Each student's name and marks

**🔹 Process:**

* Store data in array of structures
* Compare marks of all students to find the highest

**🔹 Output:**

* Display name and marks of the student with highest score

#include <stdio.h>

struct Student {

char name[50];

int marks;

};

int main() {

int i, n, maxIndex = 0;

printf("Enter number of students: ");

scanf("%d", &n);

struct Student students[n];

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

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

printf("Name: ");

scanf(" %[^\n]", students[i].name);

printf("Marks: ");

scanf("%d", &students[i].marks);

}

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

if (students[i].marks > students[maxIndex].marks) {

maxIndex = i;

}

}

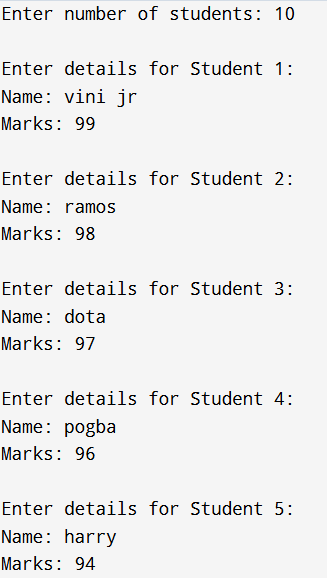
printf("\n--- Topper ---\n");

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

printf("Marks : %d\n", students[maxIndex].marks);

return 0;

}



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

**🔹 Input:**

* Number of students
* Each student's name, roll number, and marks

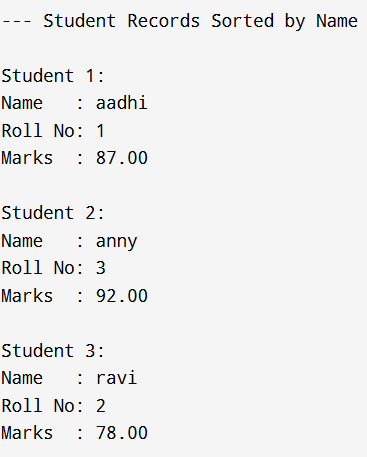
**🔹 Process:**

* Store the student details in an array of structures
* Use strcmp() in a nested loop to compare and sort names alphabetically

**🔹 Output:**

* Display the student records sorted by name

21



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

* **Input:** Integer, float, and string values one after another
* **Process:** Store each value in the union (overwriting previous)
* **Output:** Print each stored value after assignment

#include <stdio.h>

// Define a union with different data types

union Data {

int i;

float f;

char str[20];

};

int main() {

union Data data;

// Store and display an integer

data.i = 10;

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

// Store and display a float (overwrites previous data)

data.f = 220.5;

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

// Store and display a string (overwrites previous data)

// Using strcpy to assign string to union member

strcpy(data.str, "Hello");

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

return 0;

}

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

* **Input:** Values assigned to members of structure and union
* **Process:** Store and overwrite values in union; store all values in structure; print members and sizes
* **Output:** Display all structure members correctly; union only last assigned member correctly; sizes of structure and union

#include <stdio.h>

#include <string.h>

// Define structure

struct MyStruct {

int i;

float f;

char str[20];

};

// Define union

union MyUnion {

int i;

float f;

char str[20];

};

int main() {

struct MyStruct s;

union MyUnion u;

// Assign values to structure members

s.i = 10;

s.f = 220.5;

strcpy(s.str, "Structure");

// Assign values to union members (one at a time)

u.i = 10;

u.f = 220.5; // overwrites u.i

strcpy(u.str, "Union"); // overwrites u.f

// Print structure members

printf("Structure values:\n");

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

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

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

// Print union members

printf("Union values:\n");

printf("Integer: %d\n", u.i); // may print garbage, as last assignment was string

printf("Float : %.2f\n", u.f); // may print garbage

printf("String : %s\n", u.str); // valid

// Print sizes

printf("\nSize of structure: %lu bytes\n", sizeof(s));

printf("Size of union : %lu bytes\n", sizeof(u));

return 0;

}

