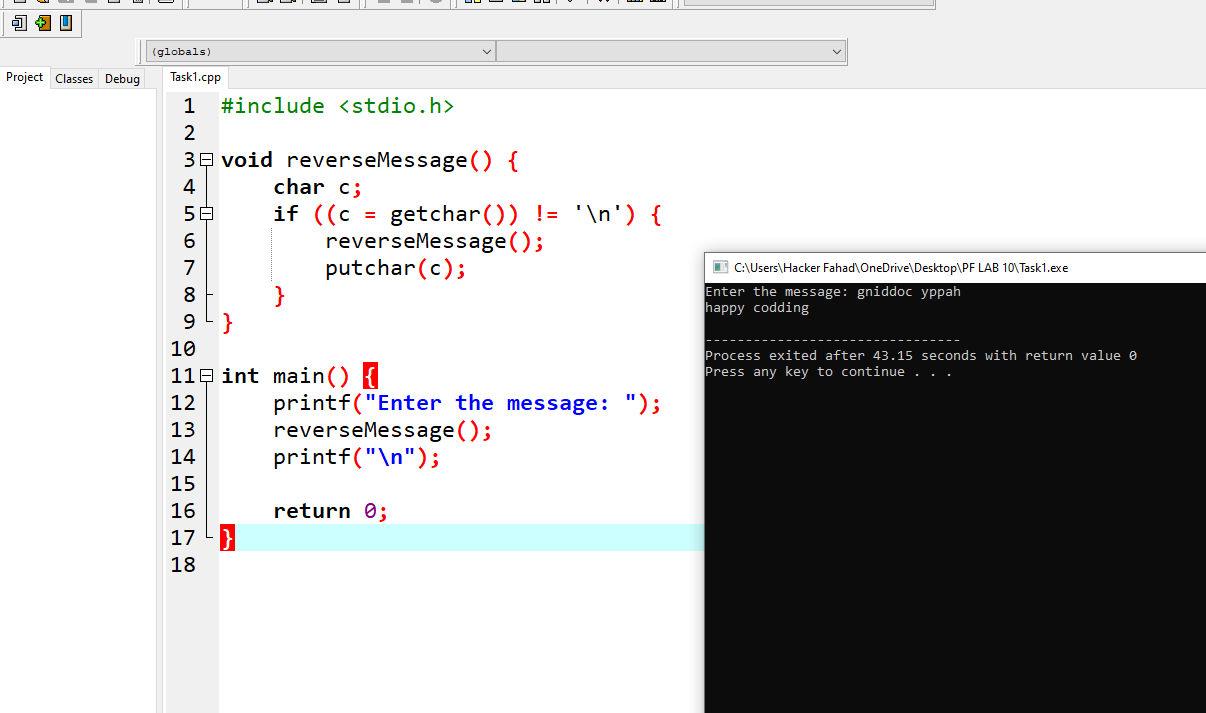
**PF LAB 10**

**TASK 1:**

****

#include <stdio.h>

void reverseMessage() {

char c;

if ((c = getchar()) != '\n') {

reverseMessage();

putchar(c);

}

}

int main() {

printf("Enter the message: ");

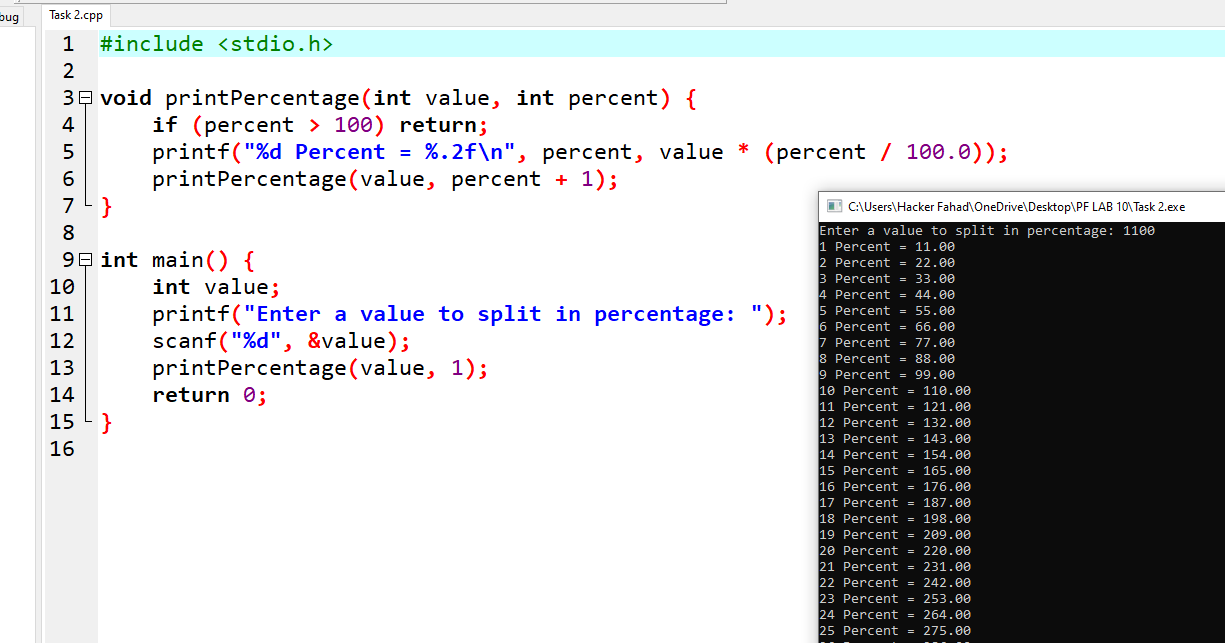
reverseMessage();

printf("\n");

return 0;

}

**TASK 2:**



#include <stdio.h>

void printPercentage(int value, int percent) {

if (percent > 100) return;

printf("%d Percent = %.2f\n", percent, value \* (percent / 100.0));

printPercentage(value, percent + 1);

}

int main() {

int value;

printf("Enter a value to split in percentage: ");

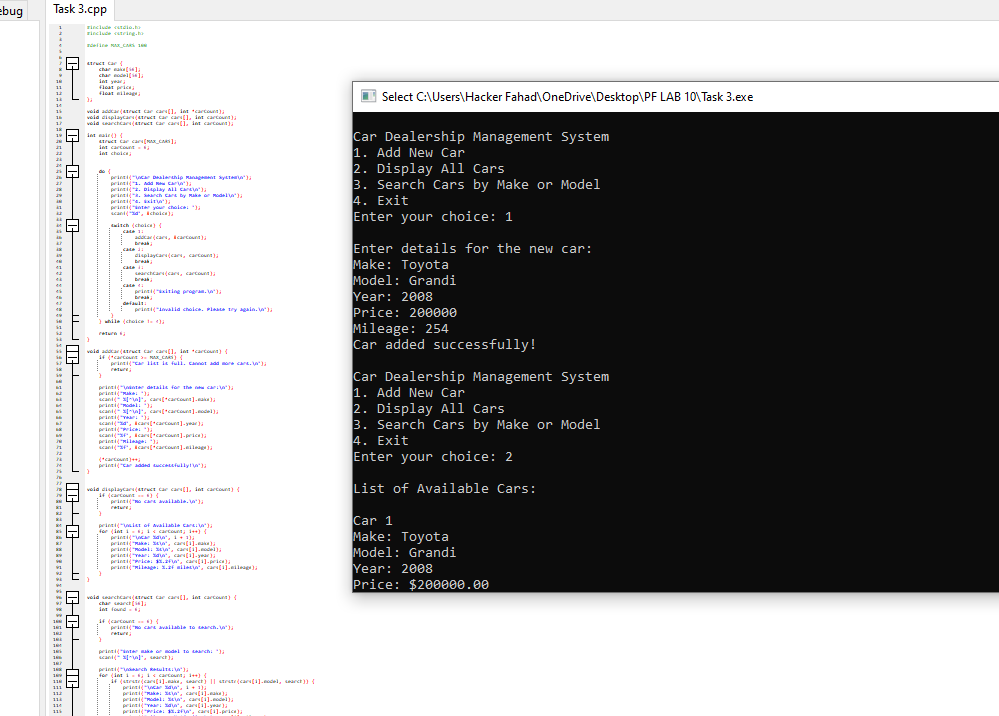
scanf("%d", &value);

printPercentage(value, 1);

return 0;

}

**TASK 3:**

****

#include <stdio.h>

#include <string.h>

#define MAX\_CARS 100

struct Car {

char make[50];

char model[50];

int year;

float price;

float mileage;

};

void addCar(struct Car cars[], int \*carCount);

void displayCars(struct Car cars[], int carCount);

void searchCars(struct Car cars[], int carCount);

int main() {

struct Car cars[MAX\_CARS];

int carCount = 0;

int choice;

do {

printf("\nCar Dealership Management System\n");

printf("1. Add New Car\n");

printf("2. Display All Cars\n");

printf("3. Search Cars by Make or Model\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addCar(cars, &carCount);

break;

case 2:

displayCars(cars, carCount);

break;

case 3:

searchCars(cars, carCount);

break;

case 4:

printf("Exiting program.\n");

break;

default:

printf("Invalid choice. Please try again.\n");

}

} while (choice != 4);

return 0;

}

void addCar(struct Car cars[], int \*carCount) {

if (\*carCount >= MAX\_CARS) {

printf("Car list is full. Cannot add more cars.\n");

return;

}

printf("\nEnter details for the new car:\n");

printf("Make: ");

scanf(" %[^\n]", cars[\*carCount].make);

printf("Model: ");

scanf(" %[^\n]", cars[\*carCount].model);

printf("Year: ");

scanf("%d", &cars[\*carCount].year);

printf("Price: ");

scanf("%f", &cars[\*carCount].price);

printf("Mileage: ");

scanf("%f", &cars[\*carCount].mileage);

(\*carCount)++;

printf("Car added successfully!\n");

}

void displayCars(struct Car cars[], int carCount) {

if (carCount == 0) {

printf("No cars available.\n");

return;

}

printf("\nList of Available Cars:\n");

for (int i = 0; i < carCount; i++) {

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

printf("Make: %s\n", cars[i].make);

printf("Model: %s\n", cars[i].model);

printf("Year: %d\n", cars[i].year);

printf("Price: $%.2f\n", cars[i].price);

printf("Mileage: %.2f miles\n", cars[i].mileage);

}

}

void searchCars(struct Car cars[], int carCount) {

char search[50];

int found = 0;

if (carCount == 0) {

printf("No cars available to search.\n");

return;

}

printf("Enter make or model to search: ");

scanf(" %[^\n]", search);

printf("\nSearch Results:\n");

for (int i = 0; i < carCount; i++) {

if (strstr(cars[i].make, search) || strstr(cars[i].model, search)) {

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

printf("Make: %s\n", cars[i].make);

printf("Model: %s\n", cars[i].model);

printf("Year: %d\n", cars[i].year);

printf("Price: $%.2f\n", cars[i].price);

printf("Mileage: %.2f miles\n", cars[i].mileage);

found = 1;

}

}

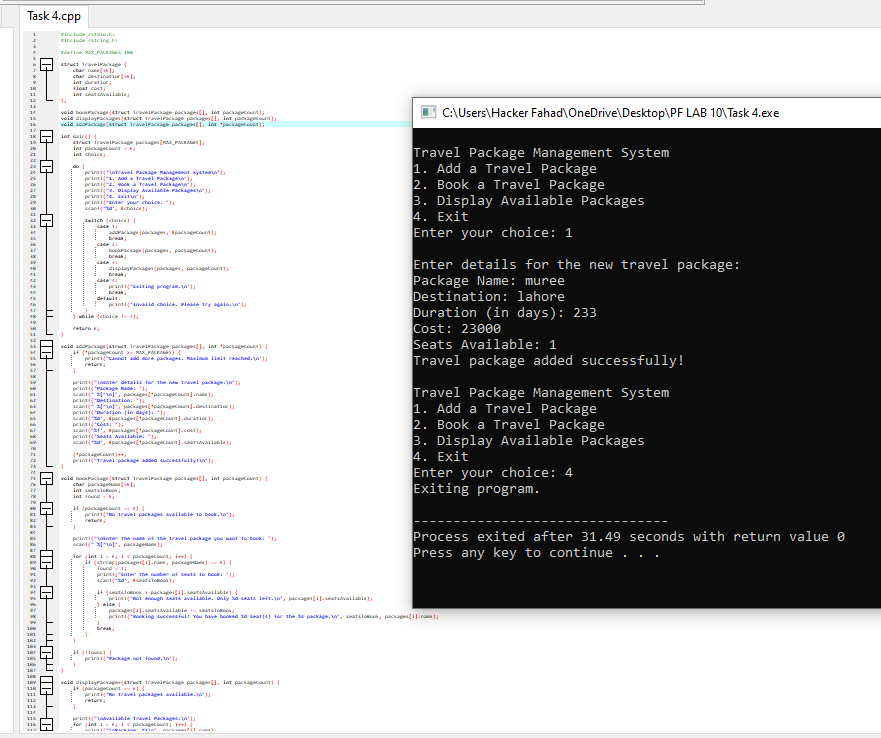
if (!found) {

printf("No cars found with make or model '%s'.\n", search);

}

}

**TASK 4:**

****

#include <stdio.h>

#include <string.h>

#define MAX\_PACKAGES 100

struct TravelPackage {

char name[50];

char destination[50];

int duration;

float cost;

int seatsAvailable;

};

void bookPackage(struct TravelPackage packages[], int packageCount);

void displayPackages(struct TravelPackage packages[], int packageCount);

void addPackage(struct TravelPackage packages[], int \*packageCount);

int main() {

struct TravelPackage packages[MAX\_PACKAGES];

int packageCount = 0;

int choice;

do {

printf("\nTravel Package Management System\n");

printf("1. Add a Travel Package\n");

printf("2. Book a Travel Package\n");

printf("3. Display Available Packages\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addPackage(packages, &packageCount);

break;

case 2:

bookPackage(packages, packageCount);

break;

case 3:

displayPackages(packages, packageCount);

break;

case 4:

printf("Exiting program.\n");

break;

default:

printf("Invalid choice. Please try again.\n");

}

} while (choice != 4);

return 0;

}

void addPackage(struct TravelPackage packages[], int \*packageCount) {

if (\*packageCount >= MAX\_PACKAGES) {

printf("Cannot add more packages. Maximum limit reached.\n");

return;

}

printf("\nEnter details for the new travel package:\n");

printf("Package Name: ");

scanf(" %[^\n]", packages[\*packageCount].name);

printf("Destination: ");

scanf(" %[^\n]", packages[\*packageCount].destination);

printf("Duration (in days): ");

scanf("%d", &packages[\*packageCount].duration);

printf("Cost: ");

scanf("%f", &packages[\*packageCount].cost);

printf("Seats Available: ");

scanf("%d", &packages[\*packageCount].seatsAvailable);

(\*packageCount)++;

printf("Travel package added successfully!\n");

}

void bookPackage(struct TravelPackage packages[], int packageCount) {

char packageName[50];

int seatsToBook;

int found = 0;

if (packageCount == 0) {

printf("No travel packages available to book.\n");

return;

}

printf("\nEnter the name of the travel package you want to book: ");

scanf(" %[^\n]", packageName);

for (int i = 0; i < packageCount; i++) {

if (strcmp(packages[i].name, packageName) == 0) {

found = 1;

printf("Enter the number of seats to book: ");

scanf("%d", &seatsToBook);

if (seatsToBook > packages[i].seatsAvailable) {

printf("Not enough seats available. Only %d seats left.\n", packages[i].seatsAvailable);

} else {

packages[i].seatsAvailable -= seatsToBook;

printf("Booking successful! You have booked %d seat(s) for the %s package.\n", seatsToBook, packages[i].name);

}

break;

}

}

if (!found) {

printf("Package not found.\n");

}

}

void displayPackages(struct TravelPackage packages[], int packageCount) {

if (packageCount == 0) {

printf("No travel packages available.\n");

return;

}

printf("\nAvailable Travel Packages:\n");

for (int i = 0; i < packageCount; i++) {

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

printf("Destination: %s\n", packages[i].destination);

printf("Duration: %d days\n", packages[i].duration);

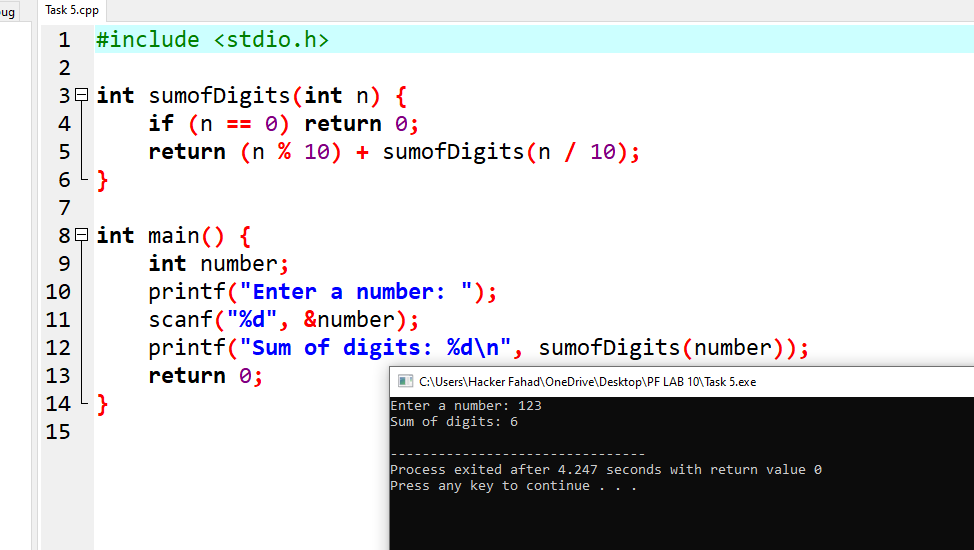
printf("Cost: $%.2f\n", packages[i].cost);

printf("Seats Available: %d\n", packages[i].seatsAvailable);

}

}

**TASK 5:**

****

#include <stdio.h>

int sumofDigits(int n) {

if (n == 0) return 0;

return (n % 10) + sumofDigits(n / 10);

}

int main() {

int number;

printf("Enter a number: ");

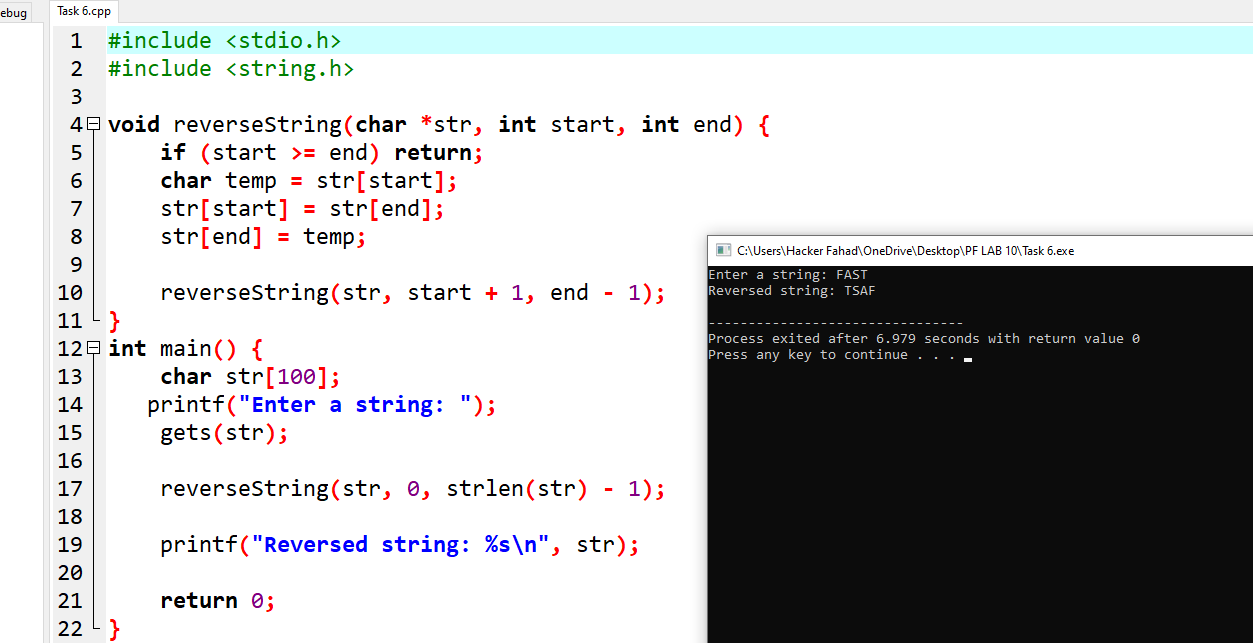
scanf("%d", &number);

printf("Sum of digits: %d\n", sumofDigits(number));

return 0;

}

**TASK 6:**

****

#include <stdio.h>

#include <string.h>

void reverseString(char \*str, int start, int end) {

if (start >= end) return;

char temp = str[start];

str[start] = str[end];

str[end] = temp;

reverseString(str, start + 1, end - 1);

}

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

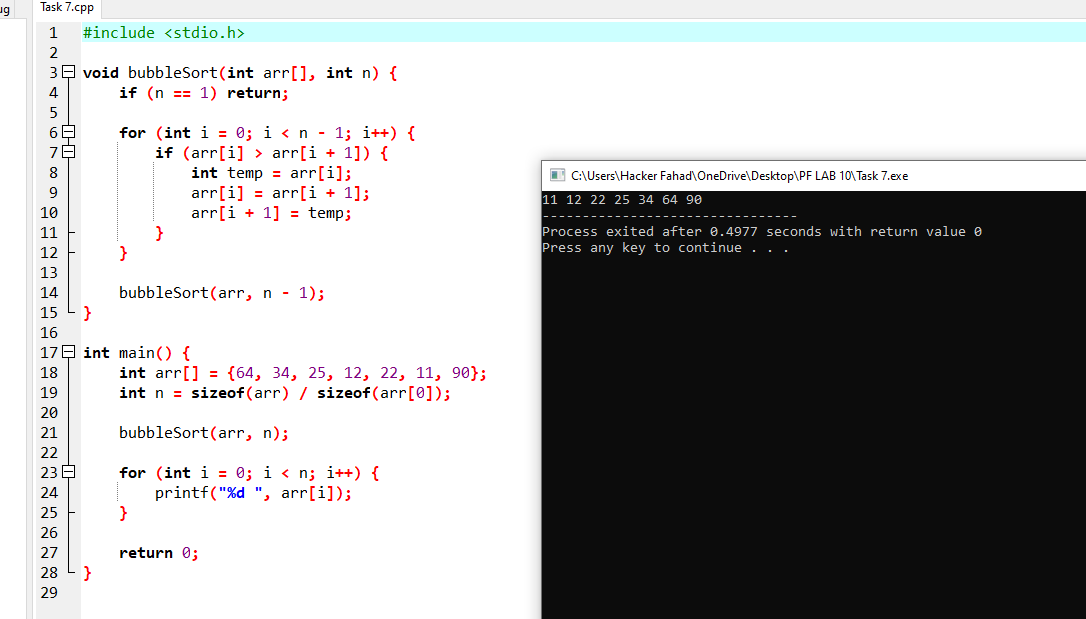
reverseString(str, 0, strlen(str) - 1);

printf("Reversed string: %s\n", str);

return 0;

}

**TASK 7:**

****

#include <stdio.h>

void bubbleSort(int arr[], int n) {

if (n == 1) return;

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

if (arr[i] > arr[i + 1]) {

int temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

}

}

bubbleSort(arr, n - 1);

}

int main() {

int arr[] = {64, 34, 25, 12, 22, 11, 90};

int n = sizeof(arr) / sizeof(arr[0]);

bubbleSort(arr, n);

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

printf("%d ", arr[i]);

}

return 0;

}

**TASK 8:**

**A screenshot of a computer program

Description automatically generated**

#include <stdio.h>

void printArray(int arr[], int n) {

if (n == 0) {

return;

}

printArray(arr, n - 1);

printf("%d ", arr[n - 1]);

}

int main() {

int arr[] = {1, 2, 3, 4, 5};

int n = sizeof(arr) / sizeof(arr[0]);

printArray(arr, n);

return 0;

}

**TASK 9:**

**A screenshot of a computer program

Description automatically generated**

#include <stdio.h>

struct Employee {

int id;

char name[50];

char department[50];

float salary;

};

int main() {

int n;

printf("Enter the number of employees: ");

scanf("%d", &n);

struct Employee employees[n];

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

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

printf("Employee ID: ");

scanf("%d", &employees[i].id);

printf("Name: ");

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

printf("Department: ");

scanf(" %[^\n]", employees[i].department);

printf("Salary: ");

scanf("%f", &employees[i].salary);

}

printf("\nEmployee Details:\n");

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

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

printf("ID: %d\n", employees[i].id);

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

printf("Department: %s\n", employees[i].department);

printf("Salary: %.2f\n", employees[i].salary);

}

return 0;

}