9. Write a program to store employee information using nested structures.

```
#include <string.h>
#define MAX_employees 100
struct date {
    int day;
    int month;
    int year;
};
struct employee {
    char name[50];
    char department[50];
    float salary;
    struct date hire_date;
};
int main(){
struct employee employees[MAX_employees];
int num_employees;
printf("Enter the number of employees:");
scanf("%d", &num_employees);
for (int i=0; i<num_employees; i++)
{
printf("Enter information for employee. %d:
i+1);
```

```
printf("Name: ");
scanf("%s", employees[i].name);
printf("Department: ");
scanf("%s", employees[i].department);
printf("Salary: ");
scanf("%f", &employees[i].salary);
printf("Hire date (dd/mm/yyyy): ");
scanf("%d %d %d", &employees[i].hire_date.
day, &employees[i].hire_date.month, &employee
[i].hire_date.year);

}
printf("\n Employee information:\n");
for(int i=0; i<num_employees; i++)
{
printf("Name: %s\n", employees[i].name);
printf("Department: %s\n", employees[i].depart-
                                    ment);
printf("Salary: %.2f\n", employees[i].salary);
printf("Hire date: %02d %02d %04d\n",
employees[i].hire_date.day, employees[i].hire_
date.month, employees[i].hire_date.year);
printf("\n");
}
return 0;
}
```

Output:

Enter number of employees: 5

Enter information for employee 1:

Name : Naveena

Department : Sales

Salary : 20000

Hire date (dd/mm/yyyy) : 06/02/2005

Enter information for employee 2:

Name: Dayanita

Department : Engineering

Salary : 500000

Hire date (dd/mm/yyyy) : 05/08/2005

Enter information for employee 3:

Name: Jayanth

Department : Manager

Salary : 200000

Hire date (dd/mm/yyyy): 06/12/2005

Enter information for employee 4:

Name : Kruthika

Department : Production

Salary : 30000

Hire date (dd/mm/yyyy): 29-08-2006

Enter information for employee 5:

Name: Fatima

Department : Human Resource

Salary: 50000

Hire date (dd/mm/yyyy): 25/10/2006

case study - Replit

replit.com/@keerthikeerthig/case-study

Gmail YouTube Maps

case study

Run

Invite Deploy

Search

Files

a.out
eminfo.c
main.c

Tools

AI Deployments Chat
Code Search Console Database
Debugger Docs Extensions
Git PostgreSQL Secrets

main.c eminfo.c

eminfo.c

```c
#include <stdio.h>
#include <string.h>
#define MAX_EMPLOYEES 100
struct date
{
    int day;
    int month;
    int year;
};
struct employee
 {
    char name[50];
    char department[50];
    float salary;
    struct date hire_date;
};

int main()
{
    struct employee employees[MAX_EMPLOYEES];
    int num_employees;

    printf("Enter the number of employees: ");
    scanf("%d", &num_employees);

    for (int i = 0; i < num_employees; i++)
    {
        printf("Enter information for employee %d:\n", i+1);

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

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

AI {-} C Ln 56, Col 2 Spaces: 2 History

Console Shell

```
~/case-study$ gcc eminfo.c
eminfo.c: In function 'main':
eminfo.c:24:5: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   24 |     scanf("%d", &num_employees);
      |     ^~~~~~~~~~~~~~~~~~~~~~~~~~~
eminfo.c:31:9: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   31 |         scanf("%s", employees[i].name);
      |         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
eminfo.c:34:9: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   34 |         scanf("%s", employees[i].department);
      |         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
eminfo.c:37:9: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   37 |         scanf("%f", &employees[i].salary);
      |         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
eminfo.c:40:9: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   40 |         scanf("%d/%d/%d", &employees[i].hire_date.day, &employees[i].hire_date.month, &employees[i].hire_date.year);
      |         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~/case-study$ ./a.out
Enter the number of employees: 5
Enter information for employee 1:
Name: NAVEENA
Department: SALES
Salary: 20000
Hire date (dd/mm/yyyy): 06/02/2005
Enter information for employee 2:
Name: DAYANITA
Department: ENGINEERING
Salary: 500000
Hire date (dd/mm/yyyy): 05/08/2005
Enter information for employee 3:
Name: JAYANTH
Department: MANAGER
Salary: 200000
Hire date (dd/mm/yyyy): 06/12/2006
Enter information for employee 4:
Name: KRITIKA
Department: PRODUCTION
Salary: 30000
Hire date (dd/mm/yyyy): 22/08/2006
Enter information for employee 5:
Name: FATIMA
```

28°C Mostly cloudy

Search

ENG IN

18:28 22-11-2023

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

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

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

        printf("Hire date (dd/mm/yyyy): ");
        scanf("%d/%d/%d", &employees[i].hire_date.day, &employees[i].hire_date.month,
&employees[i].hire_date.year);
    }

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


for (int i = 0; i < num_employees; i++)
  {
        printf("Name: %s\n", employees[i].name);
        printf("Department: %s\n", employees[i].department);
        printf("Salary: %.2f\n", employees[i].salary);
        printf("Hire date: %02d/%02d/%04d\n", employees[i].hire_date.day,
employees[i].hire_date.month, employees[i].hire_date.year);
        printf("\n");
    }

    return 0;
}
```

```
Enter information for employee 2:
Name: DAYANITA
Department: ENGINEERING
Salary: 500000
Hire date (dd/mm/yyyy): 05/08/2005
Enter information for employee 3:
Name: JAYANTH
Department: MANAGER
Salary: 200000
Hire date (dd/mm/yyyy): 06/12/2006
Enter information for employee 4:
Name: KRITIKA
Department: PRODUCTION
Salary: 30000
Hire date (dd/mm/yyyy): 22/08/2006
Enter information for employee 5:
Name: FATIMA
Department: HUMANRESOURCE
Salary: 50000
Hire date (dd/mm/yyyy): 25/10/2006

Employee Information:
Name: NAVEENA
Department: SALES
Salary: 20000.00
Hire date: 06/02/2005

Name: DAYANITA
Department: ENGINEERING
Salary: 500000.00
Hire date: 05/08/2005

Name: JAYANTH
Department: MANAGER
Salary: 200000.00
Hire date: 06/12/2006

Name: KRITIKA
Department: PRODUCTION
Salary: 30000.00
Hire date: 22/08/2006

Name: FATIMA
Department: HUMANRESOURCE
Salary: 50000.00
Hire date: 25/10/2006

~/case-study$
```

2. Program to store car information using structure

```
#include <stdio.h>
// Define the structure for storing car information
 struct car
{
 char name[50];
 char engine[20];
 char FuelType[20];
 float FuelTankCapacity;
 int SeatingCapacity;
 float CityMilage;
 };
int main()
{
// Declare an array of structures to store informat
for multiple cars struct car cars[3];

//Input car information
for (int i=0; i< 3; i++)
{
printf ("Enter information for car %d: \n", i+1);
printf ("Name : ");
scanf ("%s", cars[i].name);
//Assuming the name doesn't contain spaces
```

```
printf("Engine: ");
scanf("%s", cars[i].engine);
printf("Fuel Type: ");
scanf("%s", cars[i].FuelType);
printf("Fuel Tank Capacity: ");
scanf("%f", &cars[i].FuelTankCapacity);
printf("Seating Capacity: ");
scanf("%d", &cars[i].SeatingCapacity);
printf("City Milage: ");
scanf("%f", &cars[i].CityMilage);

}
// Display car information
printf("\n Car Information: \n");
for(int i=0; i< 3; i++)

{
printf("\nCar %d:\n", i+1);
printf("Name: %s\n", cars[i].name);
printf("Engine: %s\n", cars[i].engine);
printf("Fuel Type: %s\n", cars[i].FuelType);
printf("Fuel Tank Capacity: %.2f \n", cars[i].FuelTank-
                                        Capacity);
printf("Seating Capacity: %d \n", cars[i].Seating
                                        Capacity);
printf("City Milage: %.2f\n", cars[i].CityMilage);
}
return 0;
}
```

② Output:

Enter number of cars: 3

Enter information for car 1:

Name: Ferari

Engine: Turbo

Fuel type: Diesel

Fuel tank capacity: 2000

Seating capacity: 2

City mileage: 12

Enter information for car 2:

Name: Audi

Engine: Turbo

Fuel type: Diesel

Fuel tank capacity: 2000

Seating capacity: 5

City mileage: 15

Enter information for car 3:

Name: Lamborghini

Engine: Turbo

Fuel Tank: Diesel
Fuel tank capacity: 2000
Seating capacity: 2

City mileage: 12

case study | Run | Invite | Deploy

Search

Files
- a.out
- carinfo.c
- eminfo.c
- main.c

Tools
AI, Deployments, Chat, Code Search, Console, Database, Debugger, Docs, Extensions, Git, PostgreSQL, Secrets

main.c | eminfo.c | carinfo.c

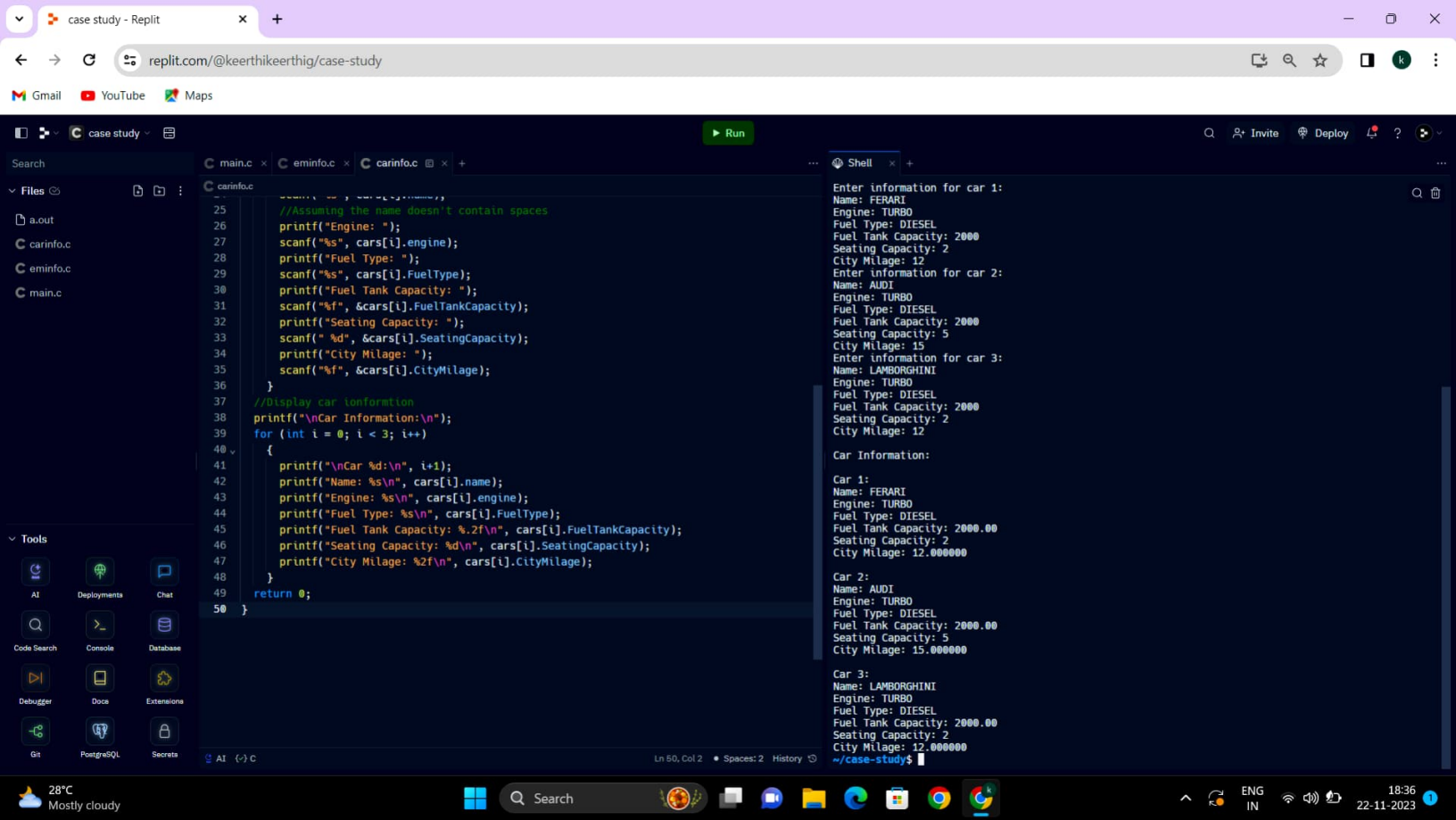
carinfo.c

```c
#include <stdio.h>

//Define the structure for storing car information
struct Car
{
  char name[50];
  char engine[20];
  char FuelType[20];
  float FuelTankCapacity;
  int SeatingCapacity;
  float CityMilage;
};

int main()
{
  //Declare an array of structures to store information for multiple cars
  struct Car cars[3];

  //Input car information
  for (int i = 0; i < 3; i++)
  {
    printf("Enter information for car %d:\n", i+1);
    printf("Name: ");
    scanf("%s", cars[i].name);
    //Assuming the name doesn't contain spaces
    printf("Engine: ");
    scanf("%s", cars[i].engine);
    printf("Fuel Type: ");
    scanf("%s", cars[i].FuelType);
    printf("Fuel Tank Capacity: ");
    scanf("%f", &cars[i].FuelTankCapacity);
    printf("Seating Capacity: ");
    scanf(" %d", &cars[i].SeatingCapacity);
    printf("City Milage: ");
```

AI | C | Ln 50, Col 2 | Spaces: 2 | History

Shell

```
~/case-study$ gcc carinfo.c
carinfo.c: In function 'main':
carinfo.c:24:7: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   24 |       scanf("%s", cars[i].name);
      |       ^~~~~~~~~~~~~~~~~~~~~~~~~
carinfo.c:27:7: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   27 |       scanf("%s", cars[i].engine);
      |       ^~~~~~~~~~~~~~~~~~~~~~~~~~~
carinfo.c:29:7: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   29 |       scanf("%s", cars[i].FuelType);
      |       ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~
carinfo.c:31:7: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   31 |       scanf("%f", &cars[i].FuelTankCapacity);
      |       ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
carinfo.c:33:7: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   33 |       scanf(" %d", &cars[i].SeatingCapacity);
      |       ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
carinfo.c:35:7: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   35 |       scanf("%f", &cars[i].CityMilage);
      |       ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~/case-study$ ./a.out
Enter information for car 1:
Name: FERARI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000
Seating Capacity: 2
City Milage: 12
Enter information for car 2:
Name: AUDI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000
Seating Capacity: 5
City Milage: 15
Enter information for car 3:
Name: LAMBORGHINI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000
Seating Capacity: 2
City Milage: 12
```


case study - Replit
replit.com/@keerthikeerthig/case-study
Gmail
YouTube
Maps
case study
Run
Invite
Deploy
Search
Files
a.out
carinfo.c
eminfo.c
main.c
Tools
AI
Deployments
Chat
Code Search
Console
Database
Debugger
Docs
Extensions
Git
PostgreSQL
Secrets
main.c
eminfo.c
carinfo.c
//Assuming the name doesn't contain spaces
printf("Engine: ");
scanf("%s", cars[i].engine);
printf("Fuel Type: ");
scanf("%s", cars[i].FuelType);
printf("Fuel Tank Capacity: ");
scanf("%f", &cars[i].FuelTankCapacity);
printf("Seating Capacity: ");
scanf(" %d", &cars[i].SeatingCapacity);
printf("City Milage: ");
scanf("%f", &cars[i].CityMilage);
}
//Display car ionformtion
printf("\nCar Information:\n");
for (int i = 0; i < 3; i++)
{
printf("\nCar %d:\n", i+1);
printf("Name: %s\n", cars[i].name);
printf("Engine: %s\n", cars[i].engine);
printf("Fuel Type: %s\n", cars[i].FuelType);
printf("Fuel Tank Capacity: %.2f\n", cars[i].FuelTankCapacity);
printf("Seating Capacity: %d\n", cars[i].SeatingCapacity);
printf("City Milage: %2f\n", cars[i].CityMilage);
}
return 0;
}
Ln 50, Col 2
Spaces: 2
History
Shell
Enter information for car 1:
Name: FERARI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000
Seating Capacity: 2
City Milage: 12
Enter information for car 2:
Name: AUDI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000
Seating Capacity: 5
City Milage: 15
Enter information for car 3:
Name: LAMBORGHINI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000
Seating Capacity: 2
City Milage: 12
Car Information:
Car 1:
Name: FERARI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000.00
Seating Capacity: 2
City Milage: 12.000000
Car 2:
Name: AUDI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000.00
Seating Capacity: 5
City Milage: 15.000000
Car 3:
Name: LAMBORGHINI
Engine: TURBO
Fuel Type: DIESEL
Fuel Tank Capacity: 2000.00
Seating Capacity: 2
City Milage: 12.000000
~/case-study$
28°C
Mostly cloudy
Search
ENG
IN
18:36
22-11-2023