SET C

Nested structure

To define a nested structure in C, we use the following syntax:

```
struct outer_structure {
  type member1;
  type member2;
  struct inner_structure {
    type inner_member1;
    type inner_member2;
  } inner;
};
```

To access the members of a nested structure, we use the dot operator (.). For example, to access inner_member1 in the nested structure above, we would use the following syntax:

```
struct outer_structure outer;
outer.inner.inner_member1 = value;
```

Program

//C program to implement the nested structure

```
#include <stdio.h>
#include <string.h>
//Declaration of the main structure
struct Organisation
{
char organisation_name[20];
char org_number[20];
// Declaration of the dependent structure
        struct Employee
        {
                int employee_id;
                char name[20];
```

```
            int salary;
        // variable is created which acts
        // as member to Organisation structure.
        } emp;
};
int main()
{
        struct Organisation org;
        strcpy(org.organisation_name,"KMCcollege");
        strcpy(org.org_number, "GFG1768");
        org.emp.employee_id = 101;
        strcpy(org.emp.name, "RAM");
        org.emp.salary = 400000;


        // Printing the details
        printf("Organisation Name : %s\n",org.organisation_name);
        printf("Organisation Number : %s\n",org.org_number);
        printf("Employee id : %d\n",org.emp.employee_id);
        printf("Employee name : %s\n",org.emp.name);
        printf("Employee Salary : %d\n",org.emp.salary);
}
```

Passing structure to a function

Following is the function declaration syntax to accept structure variable as argument.

```
returnType functionName(struct tagName argName);
```

Example:

```
void displayDetail(struct student std);
```

```
#include <stdio.h>
struct student {
  char name[50];
  int age;
};
// function prototype
void display(struct student s);
int main()
{
  struct student s1;
  printf("Enter name: ");
  scanf("%[^\n]s", s1.name);
  printf("Enter age: ");
  scanf("%d", &s1.age);
  display(s1); // passing struct as an argument
  return 0;
}
void display(struct student s)
{
  printf("\nDisplaying information\n");
  printf("Name: %s", s.name);
  printf("\nAge: %d", s.age);
}
```