It can be done in below 3 ways.

1. Passing structure by value.
2. Passing structure by address.
3. No need to pass it, Declare structure variable as global.

**Example for passing structure object by value**

1. #include<stdio.h>
3. **struct** Employee
4. {
5. **int** Id;
6. **char** Name[25];
7. **int** Age;
8. **long** Salary;
9. };
11. **void** Display(**struct** Employee);
12. **void** main()
13. {
14. **struct** Employee Emp = {1,"Kumar",29,45000};
16. Display(Emp);
18. }
20. **void** Display(**struct** Employee E)
21. {
22. printf("\n\nEmployee Id : %d",E.Id);
23. printf("\nEmployee Name : %s",E.Name);
24. printf("\nEmployee Age : %d",E.Age);
25. printf("\nEmployee Salary : %ld",E.Salary);
26. }
28. Output :
30. Employee Id : 1
31. Employee Name : Kumar
32. Employee Age : 29
33. Employee Salary : 45000

**Passing Structure by Reference**

In this approach, the reference/address structure object is passed as function argument to the definition of function.

**Example for passing structure object by reference**

1. #include<stdio.h>
3. **struct** Employee
4. {
5. **int** Id;
6. **char** Name[25];
7. **int** Age;
8. **long** Salary;
9. };
11. **void** Display(**struct** Employee\*);
12. **void** main()
13. {
14. **struct** Employee Emp = {1,"Kumar",29,45000};
16. Display(&Emp);
18. }
20. **void** Display(**struct** Employee \*E)
21. {
22. printf("\n\nEmployee Id : %d",E->Id);
23. printf("\nEmployee Name : %s",E->Name);
24. printf("\nEmployee Age : %d",E->Age);
25. printf("\nEmployee Salary : %ld",E->Salary);
26. }
28. Output :
30. Employee Id : 1
31. Employee Name : Kumar
32. Employee Age : 29
33. Employee Salary : 45000

**Function Returning Structure**

Structure is user-defined data type, like built-in data types structure can be return from function.

**Example for passing structure object by reference**

1. #include<stdio.h>
3. **struct** Employee
4. {
5. **int** Id;
6. **char** Name[25];
7. **int** Age;
8. **long** Salary;
9. };
11. Employee Input(); //Statement 1
12. **void** main()
13. {
14. **struct** Employee Emp;
16. Emp = Input();
18. printf("\n\nEmployee Id : %d",Emp.Id);
19. printf("\nEmployee Name : %s",Emp.Name);
20. printf("\nEmployee Age : %d",Emp.Age);
21. printf("\nEmployee Salary : %ld",Emp.Salary);
23. }
25. Employee Input()
26. {
27. **struct** Employee E;
29. printf("\nEnter Employee Id : ");
30. scanf("%d",&E.Id);
32. printf("\nEnter Employee Name : ");
33. scanf("%s",&E.Name);
35. printf("\nEnter Employee Age : ");
36. scanf("%d",&E.Age);
38. printf("\nEnter Employee Salary : ");
39. scanf("%ld",&E.Salary);
41. **return** E; //Statement 2
42. }
44. Output :
46. Enter Employee Id : 10
47. Enter Employee Name : Ajay
48. Enter Employee Age : 25
49. Enter Employee Salary : 15000
51. Employee Id : 10
52. Employee Name : Ajay
53. Employee Age : 25
54. Employee Salary : 15000

check the following program.

#include <stdio.h>

#include <string.h>

struct student

{

int id;

char name[20];

float percentage;

};

void func(struct student record);

int main()

{

struct student record;

[http://record.id](http://record.id/)=1;

strcpy([http://record.name](http://record.name/" \t "_blank), "Raju");

record.percentage = 86.5;

func(record);

return 0;

}

void func(struct student record)

{

printf(" Id is: %d \n", [http://record.id](http://record.id/));

printf(" Name is: %s \n", [http://record.name](http://record.name/));

printf(" Percentage is: %f \n", record.percentage);

}

#include<stdio.h>

#define MAX 3

void print(int \*);

void main()

{

int arr[3]= {1,3,6};

print(arr);

}

void print(int \*p)

{

int i;

for(i=0; i<MAX; i++)

{

printf("%d\t",\*p);

p++;

}

}

**If you want to pass an array by value then pass a structure object which has an array member to the function as argument.**

struct A {

int arr[10];

}a;

void fun ( struct A par)

{

………

}

int main ( void )

{

fun ( a );

}

**If you want to pass an array by reference then pass the array name to the function .**

void fun ( int arr[] )

{

…….

….

}

int main ( void )

{

int arr[100];

fun ( arr );

}