

```
b = &a;  
c = &b;  
printf("value of a = %d\n", a);  
printf("value of b = address of a = %u\n", b);  
printf("value of c = address of b = %u\n", c);  
printf("address of b = %u\n", c);  
printf("address of c = %u\n", &c);  
return 0;  
}
```

output →

value of a = 5
value of b = address of a = 2831685116
value of c = address of b = 2831685120
address of b = 2831685120
address of c = 2831685128

Structure :-

A structure is a composite data type that defines a grouped list of variables that are to be placed under one name in block of memory.

Program :-

structure →

struct structure-name

{

data-type member 1;

data-type member 2;

```
.  
:  
:  
data -> type member ;  
};
```

Advantages of structure :-

- It can hold variables of different data types.
- We can create objects containing different types of attributes.
- It allows us to re-use the data layout across programs.
- It is used to implement other data structure like linked list, queues, trees and graphs.

Program :-

how to use structure in program →

```
#include <stdio.h>  
#include <conio.h>  
void main ()  
{  
    struct employee  
    {  
        int id ;  
        float salary ;  
        int mobile ;  
    } ;
```



```
struct employee e1, e2, e3;  
printf ("In Enter ids, salary & mobile no. \n");  
scanf ("%d %f %d", &e1.id, &e1.salary, &e1.mobile);  
scanf ("%d %f %d", &e2.id, &e2.salary, &e2.mobile);  
printf ("%d %f %d", &e3.id, &e3.salary, &e3.mobile);  
printf ("In Entered result");  
printf ("\n %d %f %d", e1.id, e1.salary, e1.mobile);  
printf ("\n %d %f %d", e2.id, e2.salary, e2.mobile);  
printf ("\n %d %f %d", e3.id, e3.salary, e3.mobile);  
getch ();  
}
```

output →

guess the output

And write it here