

```

1  #include <stdio.h>
2
3  int main(void)
4  {
5      //variable declarations
6      int num;
7      int *ptr = NULL;
8      int **pptr = NULL; //Declaration Method 1 :- **pptr Is A Variable Of type
                           'int'
9
10     //code
11     num = 10;
12
13     printf("\n\n");
14
15     printf(" ***** BEFORE ptr = &num *****\n\n");
16     printf("Value Of 'num'           = %d\n\n", num);
17     printf("Address Of 'num'         = %p\n\n", &num);
18     printf("Value At Address Of 'num' = %d\n\n", *(&num));
19
20     //Assigning address of variable 'num' to pointer variable 'ptr'
21     //'ptr' now contains address of 'num'...hence, 'ptr' is SAME as '&num'
22     ptr = &num;
23
24     printf("\n\n");
25
26     printf(" ***** AFTER ptr = &num *****\n\n");
27     printf("Value Of 'num'           = %d\n\n", num);
28     printf("Address Of 'num'         = %p\n\n", ptr);
29     printf("Value At Address Of 'num' = %d\n\n", *ptr);
30
31     // Assigning address of variable 'ptr' to pointer-to-pointer variable
                           'pptr'
32     //'pptr' now contains the address of 'ptr' which in turn contains the
                           address of 'num'
33     // Hence, 'pptr' is SAME as '&ptr'
34     // 'ptr' is SAME as '&num'
35     // Hence, pptr = &ptr = &(&num)
36     // If ptr = &num and *ptr = *(&num) = value at address of 'num'
37     // Then, pptr = &ptr and *pptr = *(&ptr) = ptr = value at address of 'ptr'
                           i.e: 'ptr' i.e : address of 'num'
38     // Then, **pptr = **(&ptr) = *(&ptr) = *ptr = *(&num) = num = 10
39     // Hence, num = *(&num) = *ptr = *(&ptr) = **pptr
40
41     pptr = &ptr;
42
43     printf("\n\n");
44
45     printf(" ***** AFTER pptr = &ptr *****\n\n");
46     printf("Value Of 'num'           = %d\n\n", num);
47     printf("Address Of 'num' (ptr)    = %p\n\n", ptr);
48     printf("Address Of 'ptr' (pptr)   = %p\n\n", pptr);
49     printf("Value At Address Of 'ptr' (*pptr) = %p\n\n", *pptr);
50     printf("Value At Address Of 'num' (*ptr) (*pptr) = %d\n\n", **pptr);
51
52     return(0);
53 }
54

```