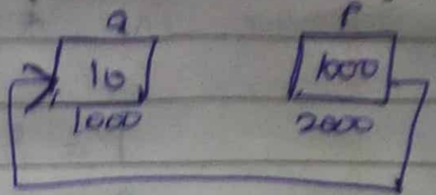
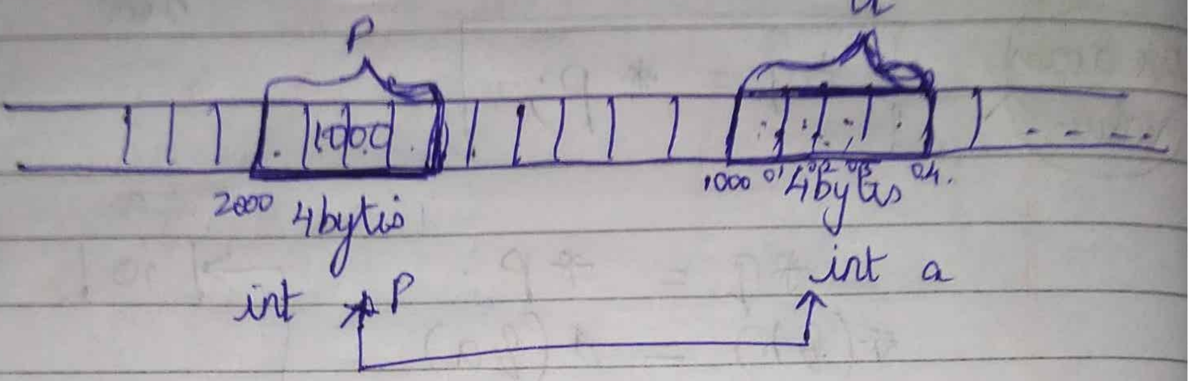
C-74 ⇒ Pointers in C

Pointer to Pointer (Double Pointer)



```
int a = 10;
int *p = &a;
```



Pointer to Variable:

Pointer is going to store address of a variable.

Double Pointer (or) Pointer to Pointer

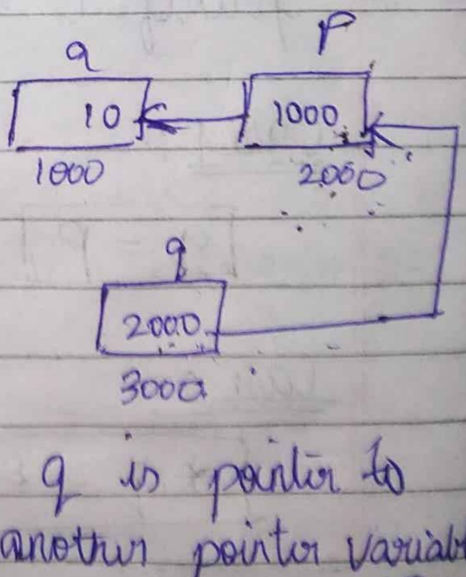
Pointer is going to store address of another pointer variable.

Example①

```
int a = 10;
int *p = &a;
```

✓ $\boxed{\text{int **q = \&p;}}$

✗ int **q = &a;



q is pointer to another pointer variable 'p'.

```
printf ("Value of a:%d", a); => 10
printf ("Value of a:%d", *p); => 10
printf ("Value of a:%d", **q); => 10.
```

Double Pointer Concept

$\Rightarrow$ ** q

$\Rightarrow$ * (*q) Value at address of p

$\Rightarrow$ * (*&p)) *(*(2000))

$\Rightarrow$ * (1000). $\Rightarrow$ Value at address of 1000.

$\Rightarrow$ [illegible] $\Rightarrow$ 10

** q

↓

** &p

↓

(* 2000)

↓

Value at 2000

↓

* (1000)

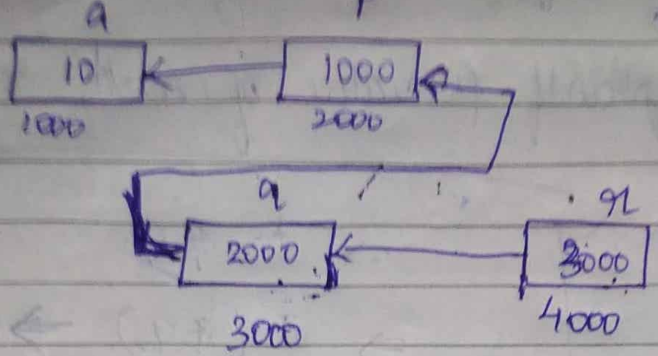
↓

Value at 100

↓

10.

* We can also increase the pointer level from double to many.



Example ②

```
int a = 10;
int *p = &a;
int **q = &P;
int ***r = &q;
printf("value of a:%d", ***r); => 10.
```

* Now, r is a 3 level pointer in which we can store address of only 2 level pointer.

* q is a 2 level pointer which can store address of a single level pointer.

* P is a 1 level pointer which can store address of only a 0 level or a variable

$$**{*}(r) \Rightarrow **(*(\&q))$$
$$\Rightarrow **(*3000)$$
$$\Rightarrow **(2000) \Longleftrightarrow *(*2000)$$
$$\Rightarrow *1000$$
$$\Rightarrow 10$$

Example ③

```
int a=10;
int *p=&a;
int **q=&p;
int ***r=&q;
printf("Value of a:%d", ***r);

    **q = 25;
printf("Value of a:%d", **q); //***r //*p.
```

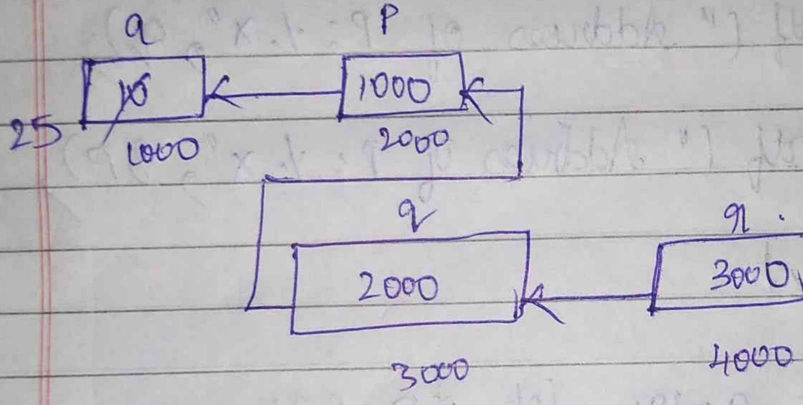
**(q)

** (&P)

** 2000

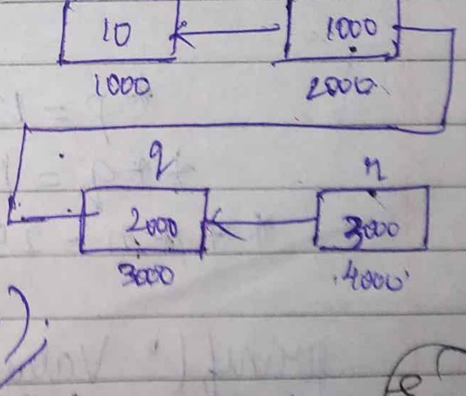
*1000 → value of address of 1000.

⇒ 25.



Example ④

```
int a=10;
int *p=&a;
int **q=&p;
int ***r=&q;
printf("Value of a:%d", *p);
```



$\boxed{*q = 25;}$ → *(&P) ⇒ *(2000) ⇒ Error

```
printf("value of a:%d" a); // *p // **q // ***r.
    ***r = 50;
printf("value of a", a);
```

Example ⑤

To print address of q;

printf (" Address of q: %x", r);
(or)
printf (" Address of q: %x", &q);

Example ⑥

To print address of P;

printf (" Address of P: %x", q);

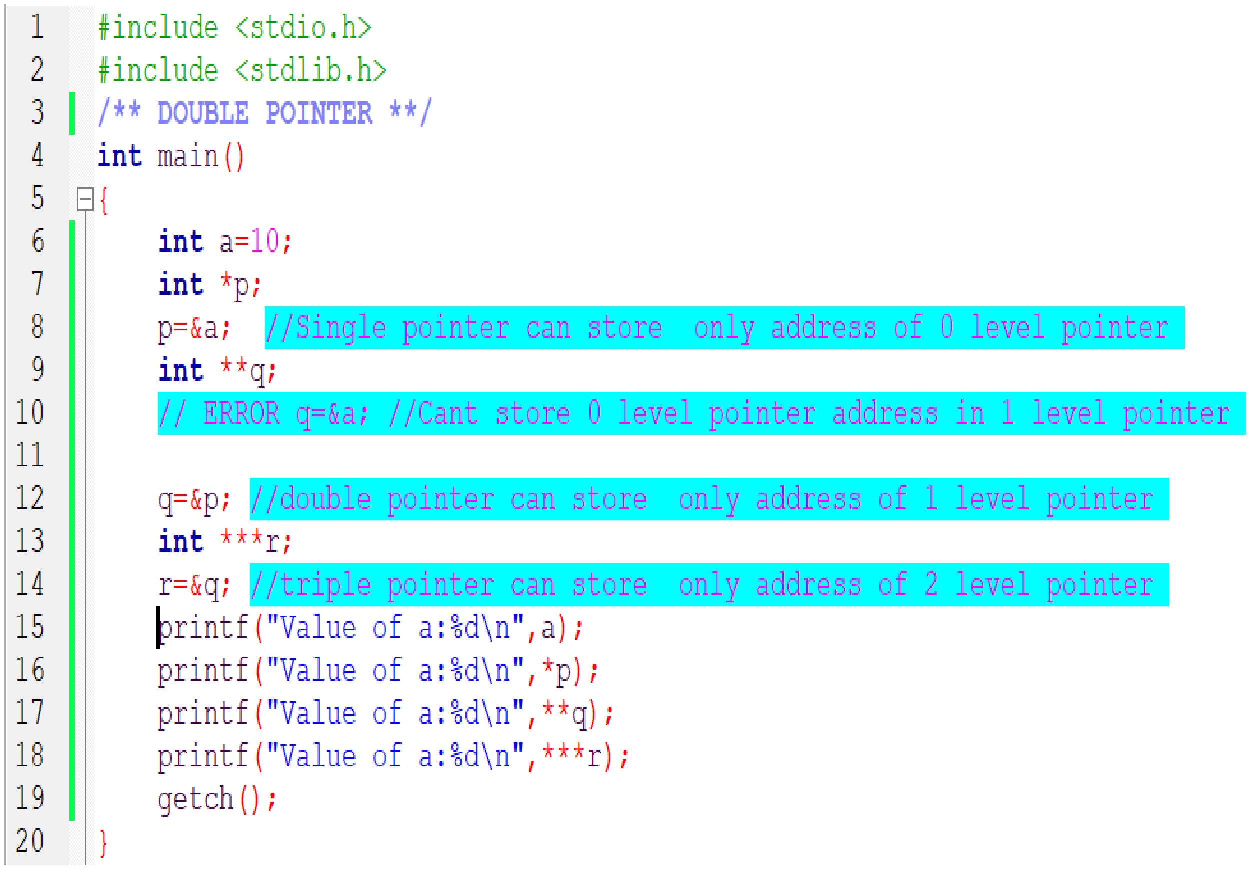
printf (" Address of P: %x", &P);

Assignment

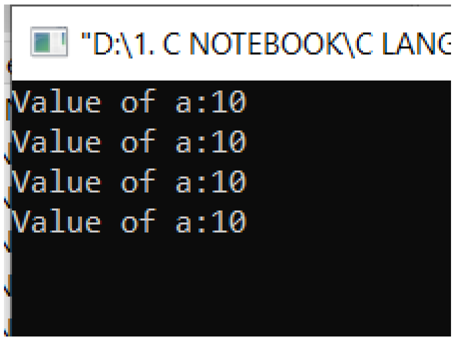
```
int a=10;  int *p=&a;
   int **q=&p;
   int ***r=&q;

     *p=12;
    **q=17;
   ***r=25;
```

printf(" Value of a": %d", ***r);

CODE 1:



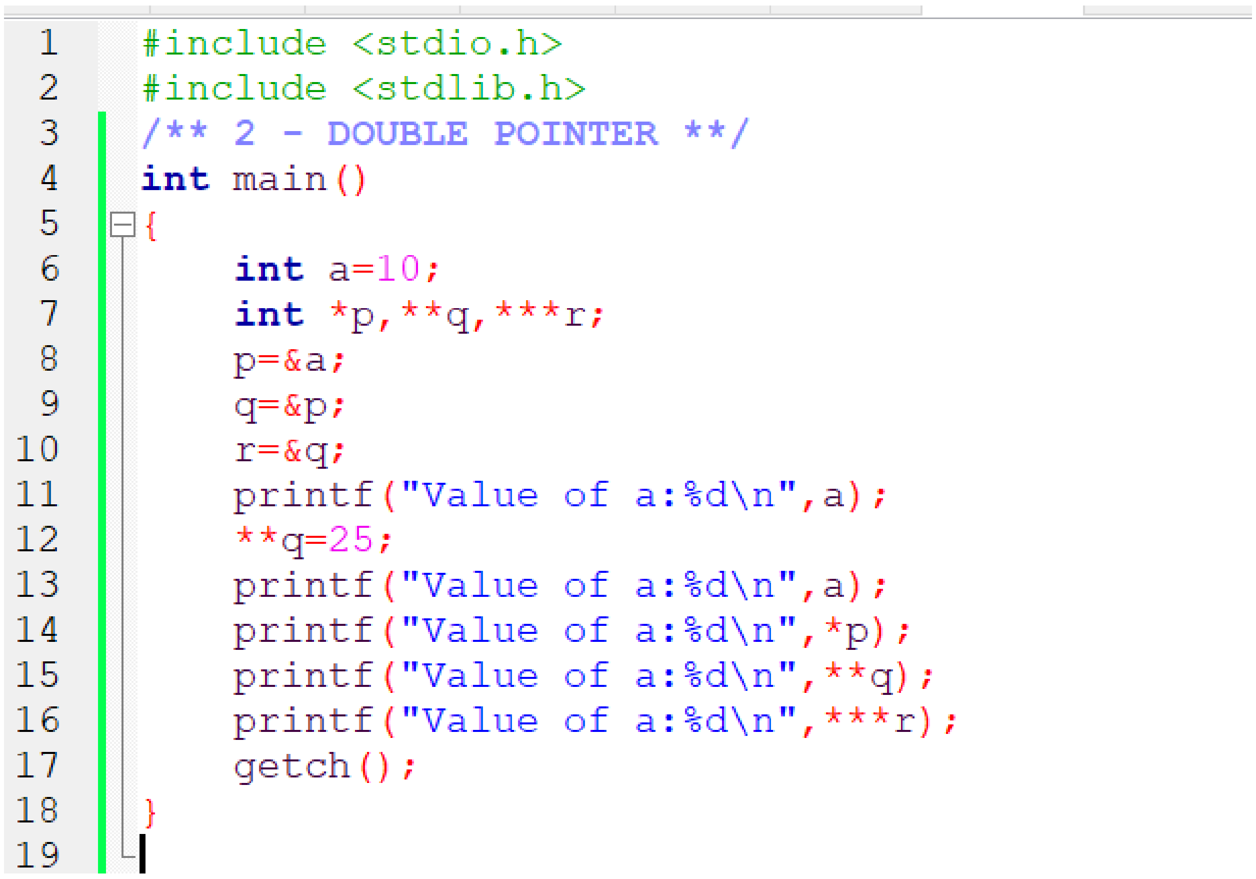
```c
#include <stdio.h>
#include <stdlib.h>
/** DOUBLE POINTER **/
int main()
{
    int a=10;
    int *p;
    p=&a;  //Single pointer can store  only address of 0 level pointer
    int **q;
    // ERROR q=&a; //Cant store 0 level pointer address in 1 level pointer

    q=&p; //double pointer can store  only address of 1 level pointer
    int ***r;
    r=&q; //triple pointer can store  only address of 2 level pointer
    printf("Value of a:%d\n",a);
    printf("Value of a:%d\n",*p);
    printf("Value of a:%d\n",**q);
    printf("Value of a:%d\n",***r);
    getch();
}
```



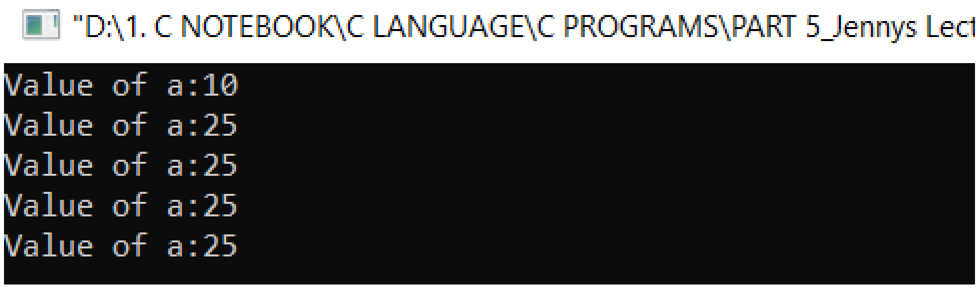
"D:\1. C NOTEBOOK\C LANG

```
Value of a:10
Value of a:10
Value of a:10
Value of a:10
```

CODE 2:



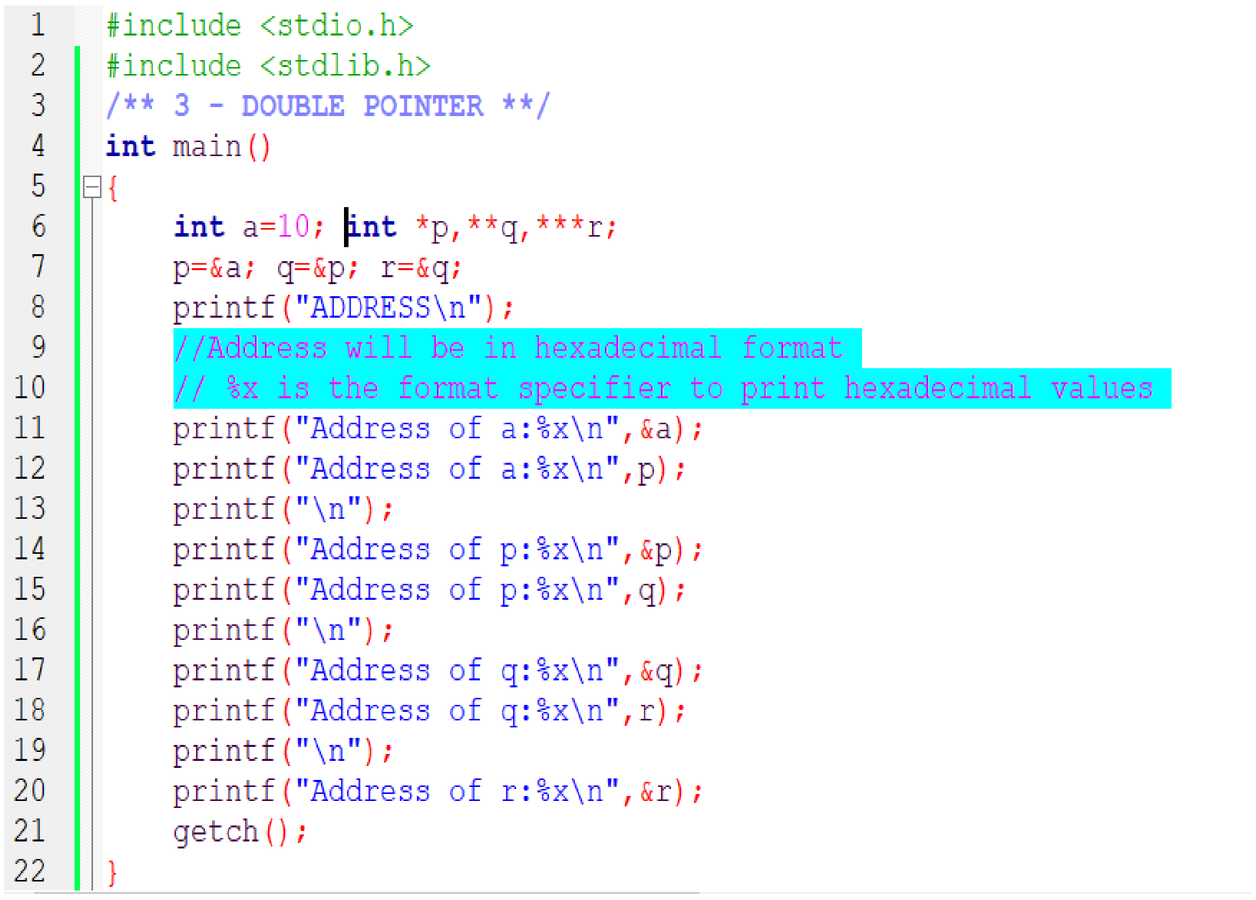
```c
#include <stdio.h>
#include <stdlib.h>
/** 2 - DOUBLE POINTER **/
int main()
{
    int a=10;
    int *p,**q,***r;
    p=&a;
    q=&p;
    r=&q;
    printf("Value of a:%d\n",a);
    **q=25;
    printf("Value of a:%d\n",a);
    printf("Value of a:%d\n",*p);
    printf("Value of a:%d\n",**q);
    printf("Value of a:%d\n",***r);
    getch();
}
```



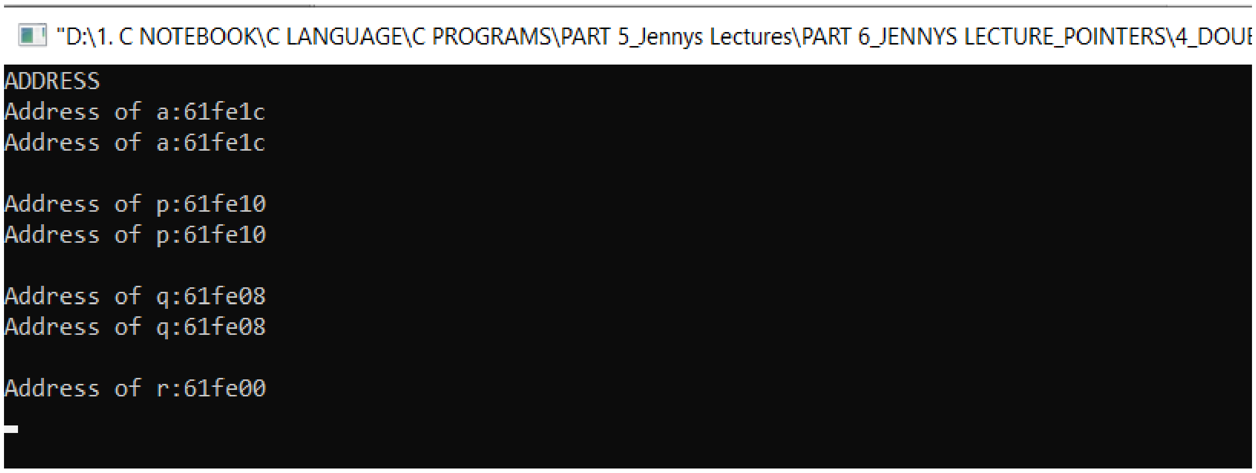
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lect

```
Value of a:10
Value of a:25
Value of a:25
Value of a:25
Value of a:25
```

CODE 3:



```c
#include <stdio.h>
#include <stdlib.h>
/** 3 - DOUBLE POINTER **/
int main()
{
    int a=10; int *p,**q,***r;
    p=&a; q=&p; r=&q;
    printf("ADDRESS\n");
    //Address will be in hexadecimal format
    // %x is the format specifier to print hexadecimal values
    printf("Address of a:%x\n",&a);
    printf("Address of a:%x\n",p);
    printf("\n");
    printf("Address of p:%x\n",&p);
    printf("Address of p:%x\n",q);
    printf("\n");
    printf("Address of q:%x\n",&q);
    printf("Address of q:%x\n",r);
    printf("\n");
    printf("Address of r:%x\n",&r);
    getch();
}
```



"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LECTURE_POINTERS\4_DOUE

```
ADDRESS
Address of a:61fe1c
Address of a:61fe1c

Address of p:61fe10
Address of p:61fe10

Address of q:61fe08
Address of q:61fe08

Address of r:61fe00
```