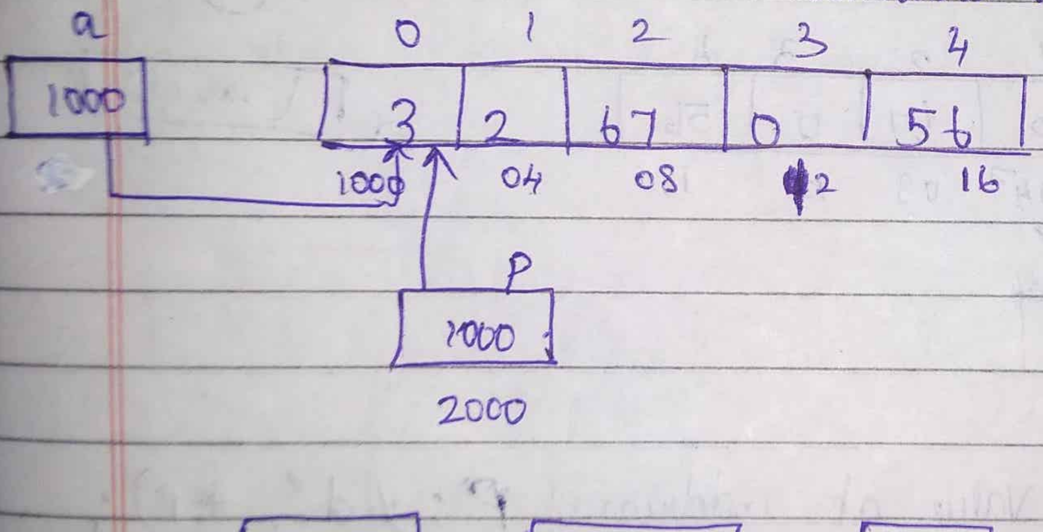
C-77 ⇒ Pointers in C ⇒ Pointer Arithmetic

Increment / Decrement with program

```
int a[] = {3, 2, 67, 0, 56}
 int *P;
```

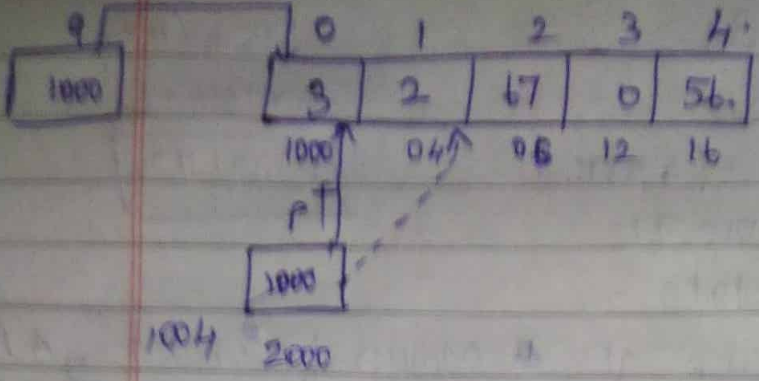
$\boxed{P = a;}$ ✓

$\boxed{a = P}$ ✗ → a is constant pointer which contains only base address.



① $\boxed{P++;}$ <=> $\boxed{P = P+1}$; $\boxed{P+1}$ ✗ Different.

Example ①:- P++;



P++ ⇒ P = P+1

⇒ 1000 + (1×4)

⇒ 1004.

Example ②:· *P;

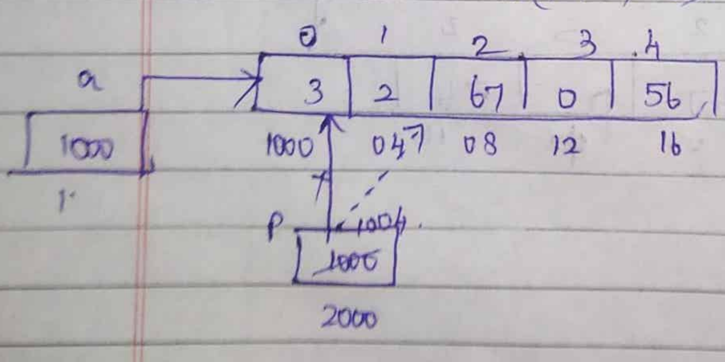
printf (" Value at address of P: %d ", *P);

output ⇒ ②

Example ③:· *P++; POST INCREMENT

printf (" Value at address of P: %d ", *P++);

*(P++) ⇒ *(1000) ⇒ ③ output.



[(1000+1)]

printf(" Value at address of P : %d ", *P);

→ ② output

Example ④: *++P;   PRE INCREMENT

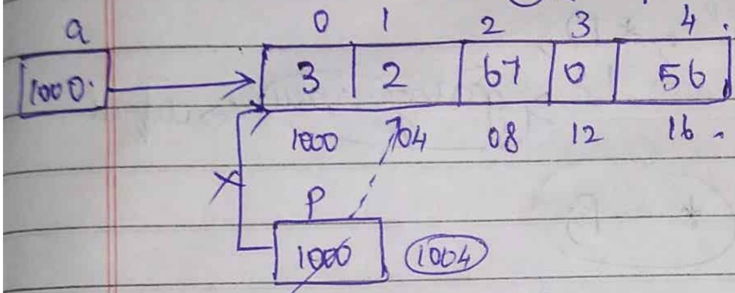
printf ("value at address of P: %d", *++P);

Note:
Associativity ⇒ Right to Left.

*++P;
← Right to left

(++P) ⇒ ~~~~(1000+1)
⇒ *(1004).
⇒ ④ ← output

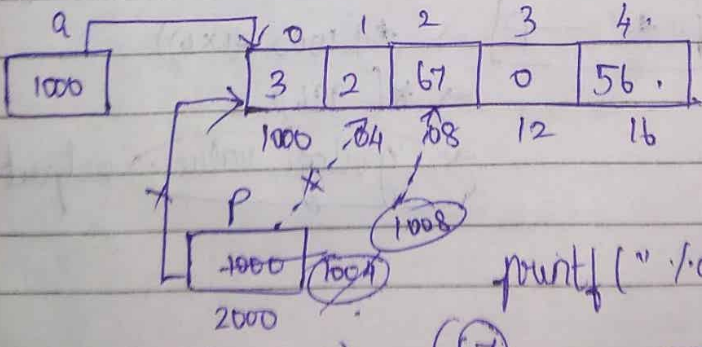
1000 + 1



Example ⑤:

printf(" %d %d", *P++, *P++);

Note: Associativity is Right to Left



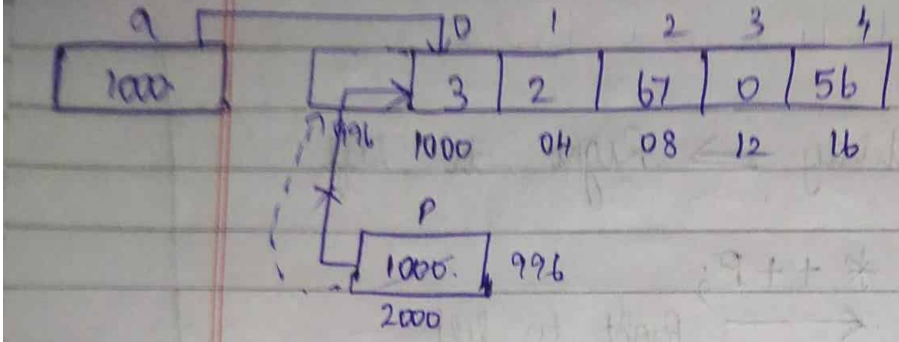
printf(" %d %d", *P++, *P++)
*1004, *1000

printf("%d", *P) ⇒ 67

Output ⇒ ② . ③

Example 6 :- *P--; POST DECREMENT

printf("%d", *P--);



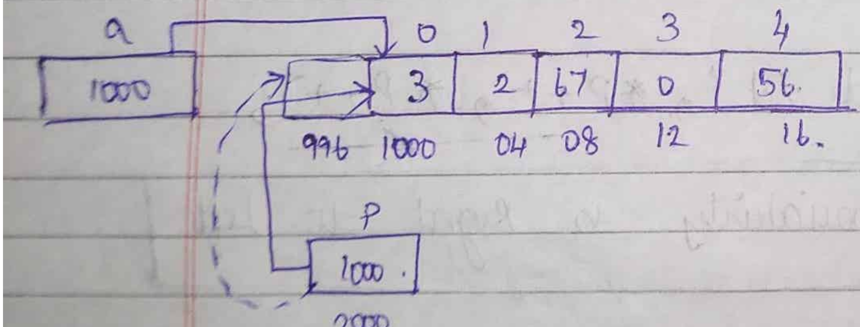
$\boxed{*P--}$ => *(1000)
=> ③ output

1000 -(1×4)
1000 - 4
996

printf("%d", *P);
└→ garbage value => output

Example 7 :- *--P;

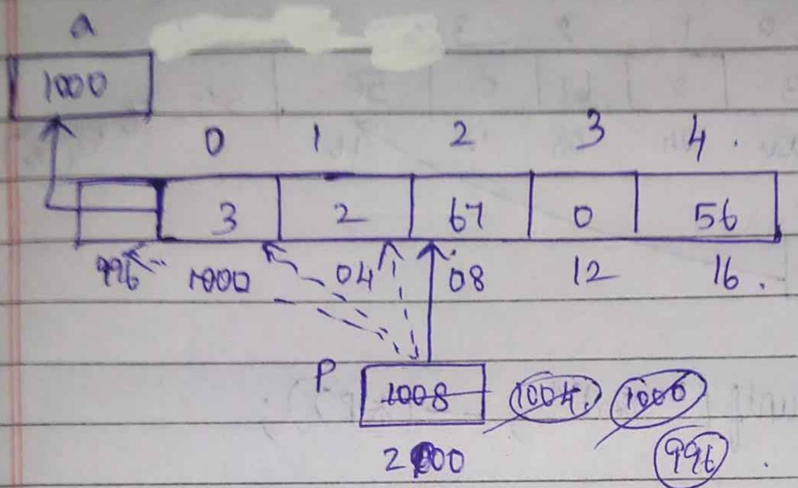
printf("%d", *--P); PRE DECREMENT



$\boxed{* --P}$ => *(1000 -(1×4))
=> *(996) .
=> garbage value output .

Example 8:
```
int a[] = {3, 2, 67, 0, 56};
int *P = &a[2];
            ③       ②       ①
printf("%d %d %d", *--P, *--P, *--P);
```

← Associativity ⇒ Right to Left



① * --P ⇒ *(1008 −(1*4))
⇒ *1004.
⇒ ② ⇒ output

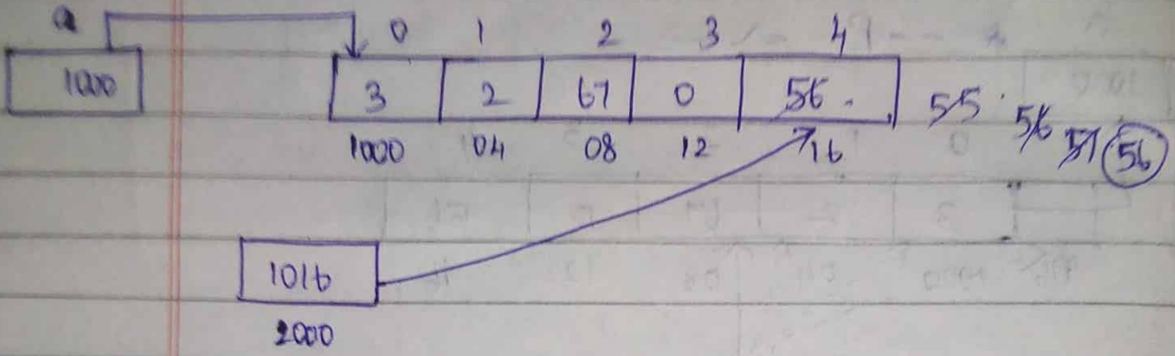
② * --P ⇒ *(1004 −(1×4))
⇒ * 1000 ⇒ ③ ⇒ output

③ * --P ⇒ *(1000 −(1×4))
⇒ * 996. ⇒ garbage value ⇒ output

```
printf("%d %d %d", *--P, *--P, *--P);
```
↳ 2 3 garbage value

Example 9 :-

```
int a[] = {3,2,67,0,56};
int *P = &a[4];
```



① printf("%d", --(*P));

⇒ --(*1016);

⇒ --(56); ⇒ 55

② printf("%d", (*P)++);

⇒ (*1016)++;

⇒ 55++; ⇒ 55

③ printf("%d", ++(*P));

⇒ ++(*1016);

⇒ ++ 56

⇒ 57

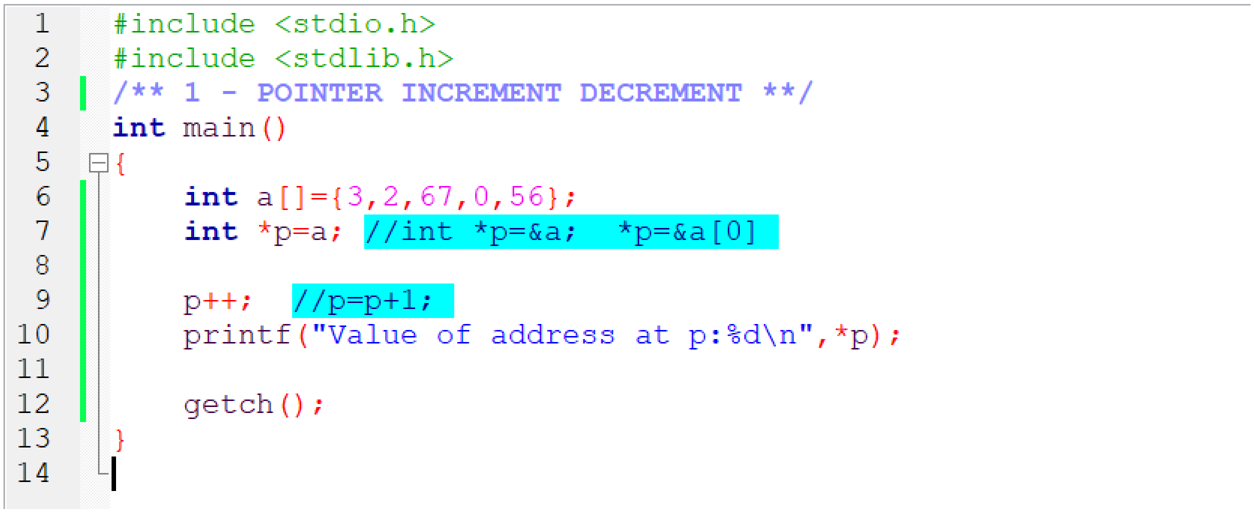
④ printf("%d", (*P)--);

⇒ (*1016)--

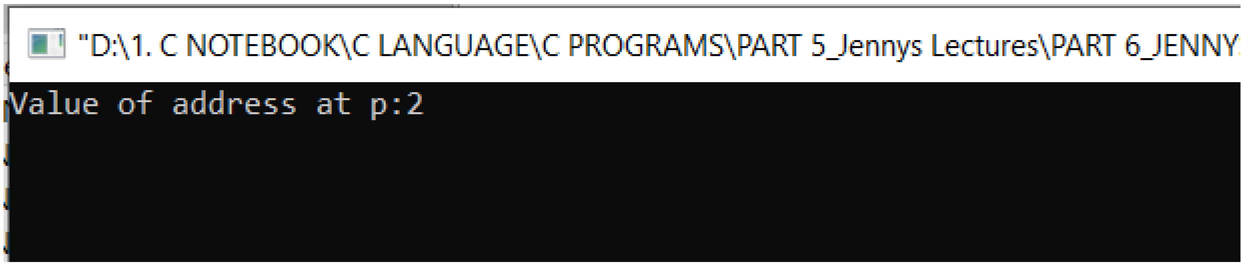
⇒ 57--

⇒ 57

CODE 1:



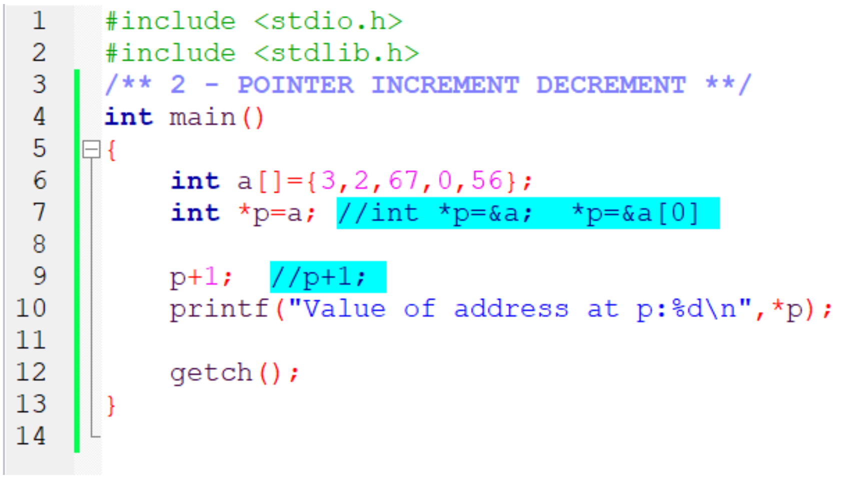
```c
#include <stdio.h>
#include <stdlib.h>
/** 1 - POINTER INCREMENT DECREMENT **/
int main()
{
    int a[]={3,2,67,0,56};
    int *p=a; //int *p=&a;   *p=&a[0]

    p++;  //p=p+1;
    printf("Value of address at p:%d\n",*p);

    getch();
}

```



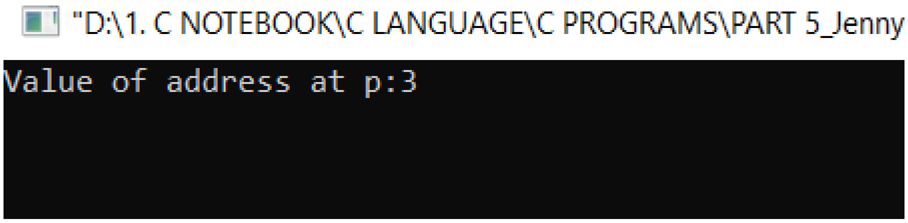
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNY

```
Value of address at p:2
```

CODE 2:

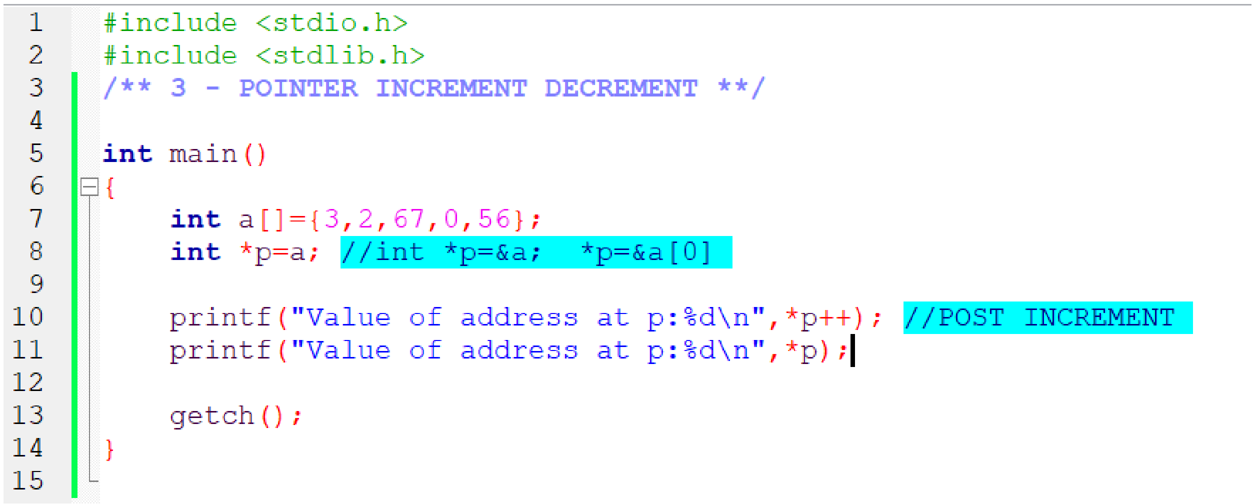


```c
#include <stdio.h>
#include <stdlib.h>
/** 2 - POINTER INCREMENT DECREMENT **/
int main()
{
    int a[]={3,2,67,0,56};
    int *p=a; //int *p=&a;   *p=&a[0]

    p+1;  //p+1;
    printf("Value of address at p:%d\n",*p);

    getch();
}

```

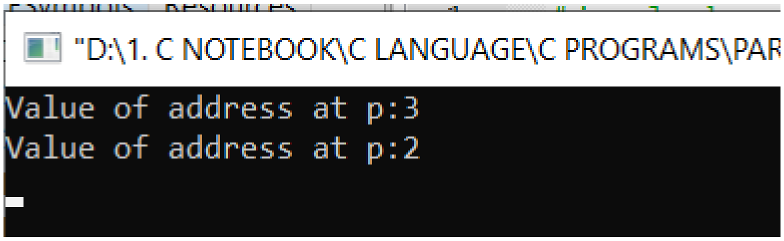


```
Value of address at p:3
```

CODE 3:

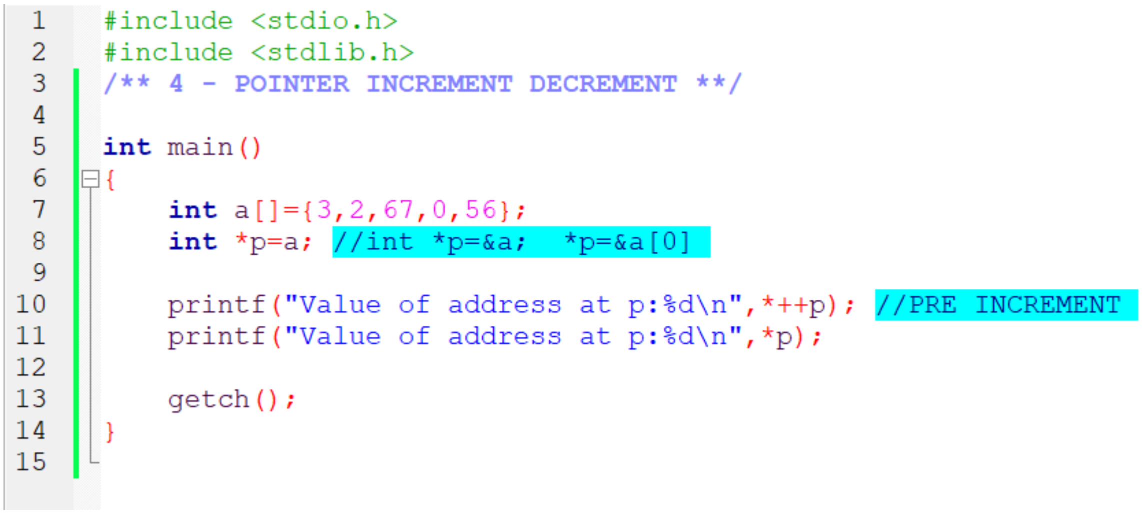


```c
#include <stdio.h>
#include <stdlib.h>
/** 3 - POINTER INCREMENT DECREMENT **/

int main()
{
    int a[]={3,2,67,0,56};
    int *p=a; //int *p=&a;   *p=&a[0]

    printf("Value of address at p:%d\n",*p++); //POST INCREMENT
    printf("Value of address at p:%d\n",*p);

    getch();
}

```

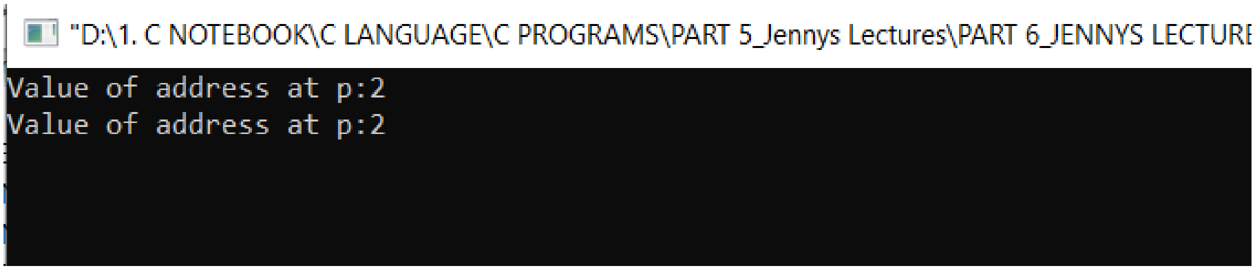


```
Value of address at p:3
Value of address at p:2
```

CODE 4:



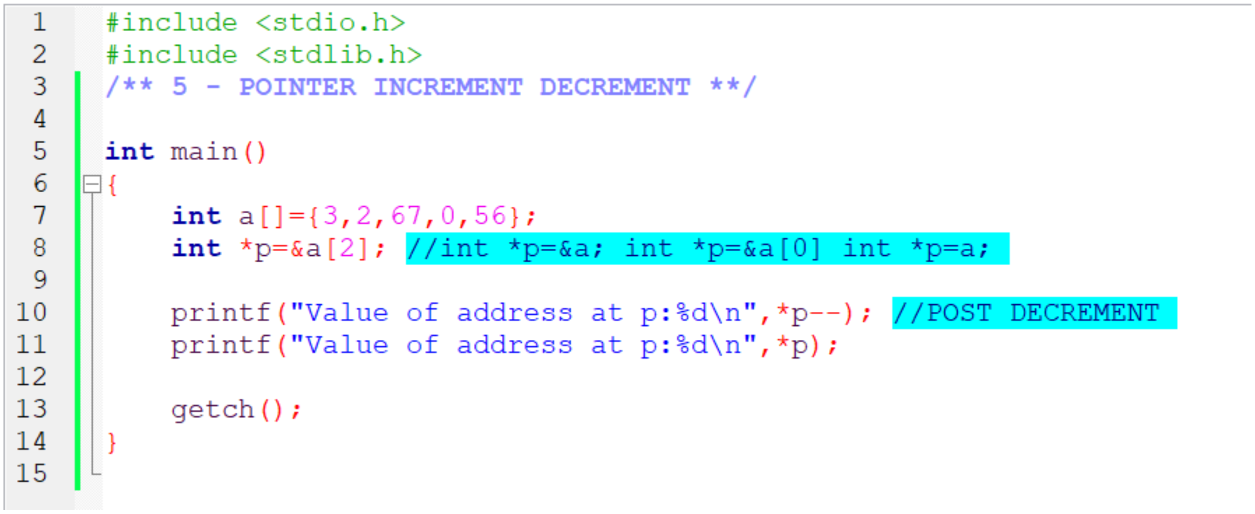
```c
#include <stdio.h>
#include <stdlib.h>
/** 4 - POINTER INCREMENT DECREMENT **/

int main()
{
    int a[]={3,2,67,0,56};
    int *p=a; //int *p=&a;   *p=&a[0]

    printf("Value of address at p:%d\n",*++p); //PRE INCREMENT
    printf("Value of address at p:%d\n",*p);

    getch();
}
```



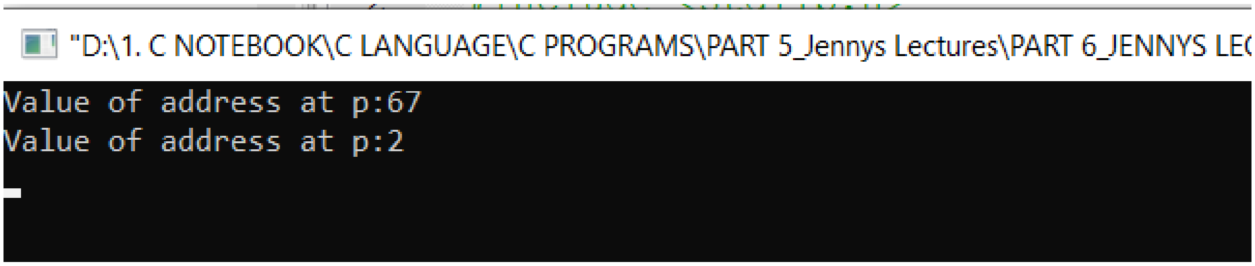
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LECTUR

```
Value of address at p:2
Value of address at p:2
```

CODE 5:



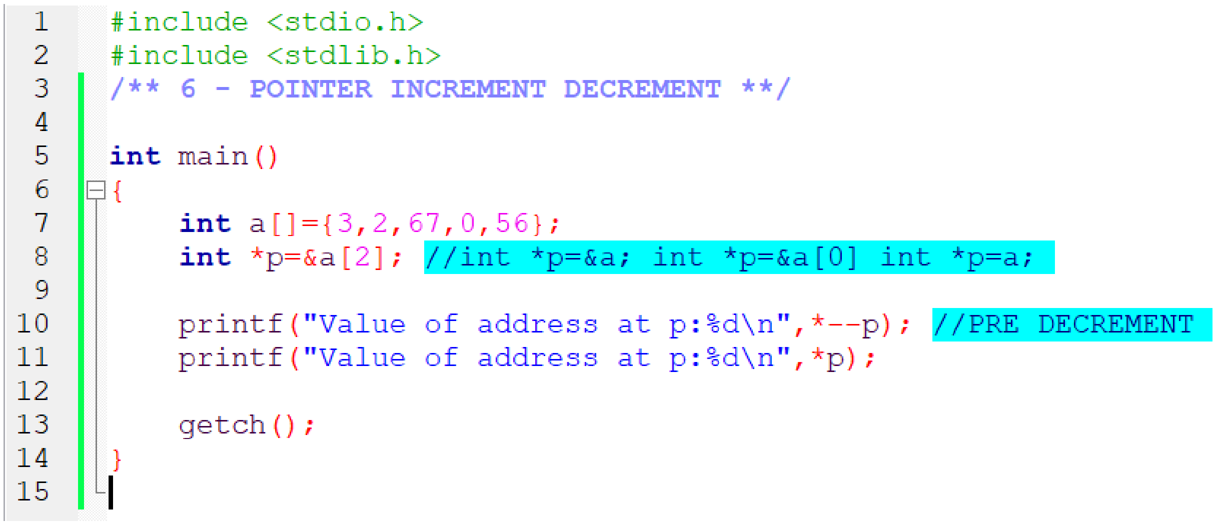
```c
#include <stdio.h>
#include <stdlib.h>
/** 5 - POINTER INCREMENT DECREMENT **/

int main()
{
    int a[]={3,2,67,0,56};
    int *p=&a[2]; //int *p=&a; int *p=&a[0] int *p=a;

    printf("Value of address at p:%d\n",*p--); //POST DECREMENT
    printf("Value of address at p:%d\n",*p);

    getch();
}
```



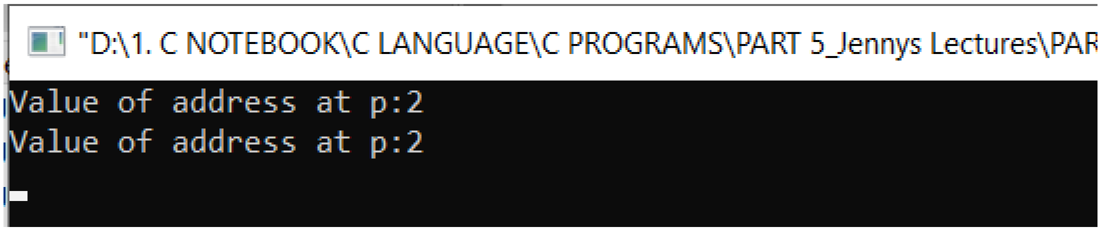
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LE

```
Value of address at p:67
Value of address at p:2
```

CODE 6:



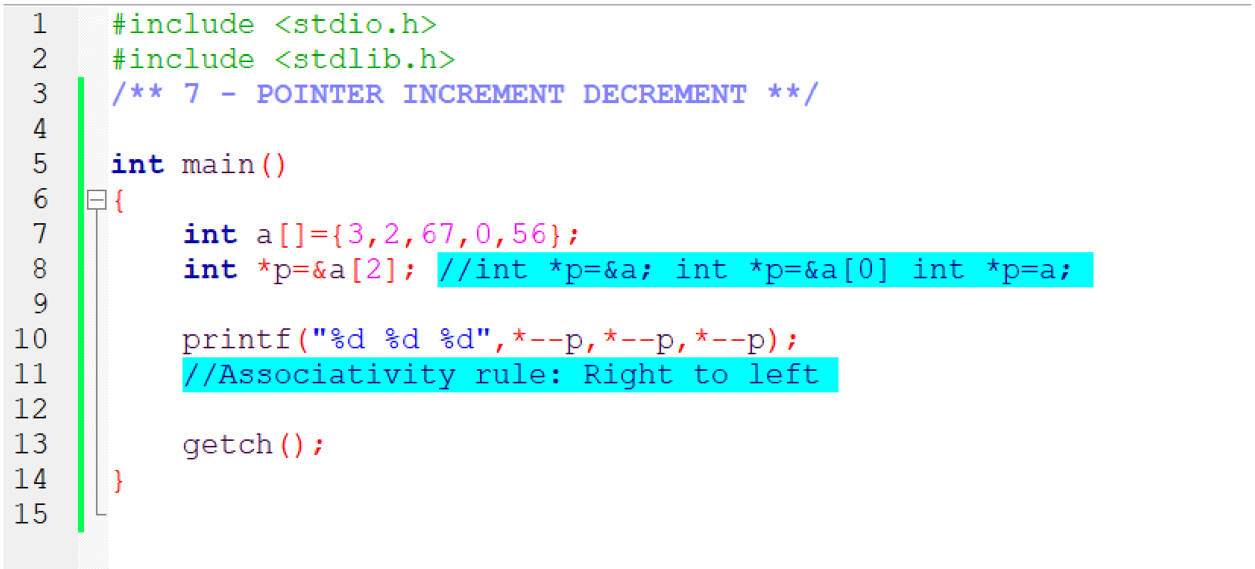
```c
#include <stdio.h>
#include <stdlib.h>
/** 6 - POINTER INCREMENT DECREMENT **/

int main()
{
    int a[]={3,2,67,0,56};
    int *p=&a[2]; //int *p=&a; int *p=&a[0] int *p=a;

    printf("Value of address at p:%d\n",*--p); //PRE DECREMENT
    printf("Value of address at p:%d\n",*p);

    getch();
}
```



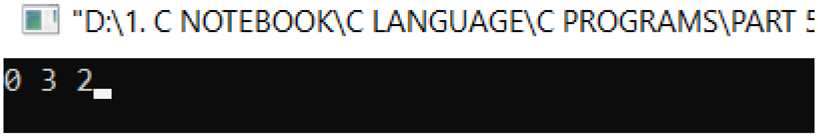
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PAR

```
Value of address at p:2
Value of address at p:2
```

CODE 7:



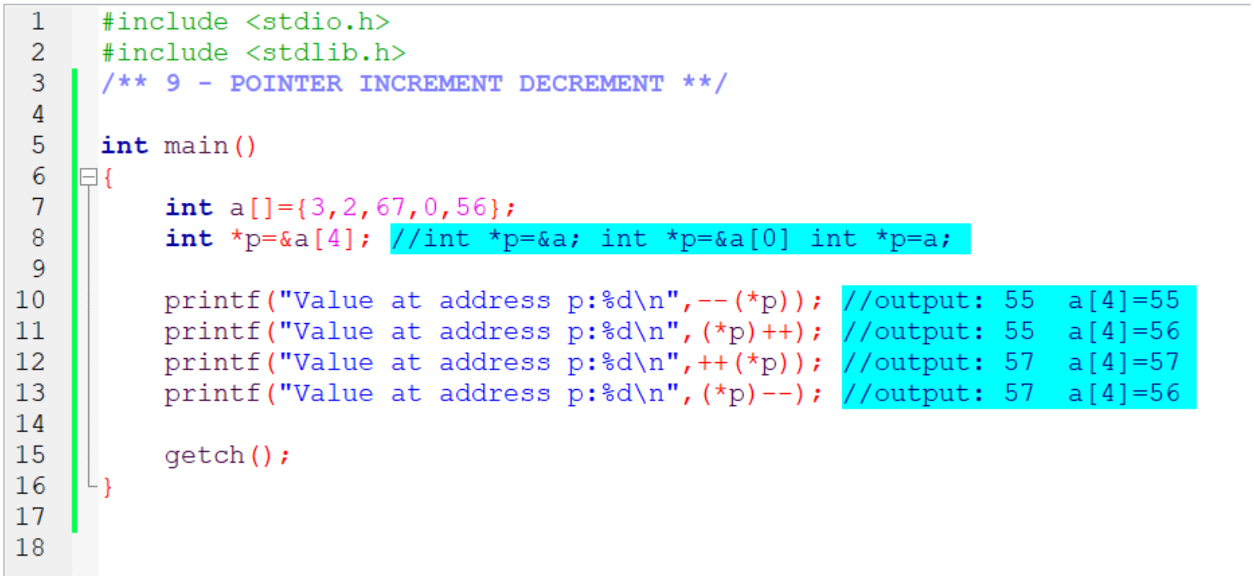
```c
#include <stdio.h>
#include <stdlib.h>
/** 7 - POINTER INCREMENT DECREMENT **/

int main()
{
    int a[]={3,2,67,0,56};
    int *p=&a[2]; //int *p=&a; int *p=&a[0] int *p=a;

    printf("%d %d %d",*--p,*--p,*--p);
    //Associativity rule: Right to left

    getch();
}

```



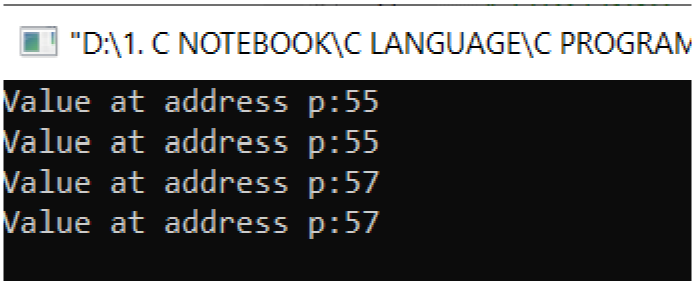
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5

```
0 3 2_
```

CODE 8:



```c
#include <stdio.h>
#include <stdlib.h>
/** 9 - POINTER INCREMENT DECREMENT **/

int main()
{
    int a[]={3,2,67,0,56};
    int *p=&a[4]; //int *p=&a; int *p=&a[0] int *p=a;

    printf("Value at address p:%d\n",--(*p)); //output: 55  a[4]=55
    printf("Value at address p:%d\n",(*p)++); //output: 55  a[4]=56
    printf("Value at address p:%d\n",++(*p)); //output: 57  a[4]=57
    printf("Value at address p:%d\n",(*p)--); //output: 57  a[4]=56

    getch();
}


```



"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAM

```
Value at address p:55
Value at address p:55
Value at address p:57
Value at address p:57
```