# CS & IT ENGINEERING

## Programming in C

**Arrays and Pointers**

**Lec- 07**

By- Pankaj Sharma sir


Arrays and Pointers-7

Q

```
int a[4] = {10, 20, 30, 40};
int *P[4] = {a+2, a, a+1, a+3};
int **q;
q = &P[0];
pf("%d", *++*++q);
```

→ 20

| a[0] | a[1] | a[2] | a[3] |
|---|---|---|---|
| 10 | 20 | 30 | 40 |

| &a[2] | &a[0] (&a[1]) | &a[1] | &a[3] |
|---|---|---|---|
| P[0] | P[1] | P[2] | P[3] |

$**q$

$\cancel{**\&}P[1]$

$*P[1]$

$\cancel{*\&}a[1] = a[1]$

$*++*(++q)$

q

$\boxed{\cancel{\&P[0]}\ \&P[1]}$

(i) $q = q+1$

(ii) $*++*q$

$*(++(*q))$

(i) $*q = *q + 1 \Rightarrow P[1] = P[1] + 1$

$= \&a[0] + 1 = \&a[1]$

(ii) $**q$

$q = \&P[1]$

$*q = *\&P[1] = P[1]$

Q

```
int a[4] = {10,20,30,40};
int *p[4] = {a+2, a, a+1, a+3};
int **q;
q = &p[0];

*++*++q;
--q;
printf("%d", *--*++q);
```

| a[0] | a[1] | a[2] | a[3] |
|---|---|---|---|
| 10 | 20 | 30 | 40 |

| | | | |
|---|---|---|---|
| &a[2] | ~~&a[0]~~ &a[1] | &a[1] | &a[3] |
| p[0] | p[1] | p[2] | p[3] |

q ~~&p[0]~~ ~~&p[1]~~ ~~&p[0]~~ &p[1]

p[1] = p[1] + 1
&a[0] + 1

10

(i) *++*(++q)

a) q = q + 1
b) *++*q

*++(*q)

a) *q = *q + 1
b) **q → meaningless

--q ⇒ q = q - 1
= &p[1] - 1 = &p[0]

*--*(++q)

a) q = q + 1
b) *--*q

q = &p[0] + 1 = &p[1]

*--(*q)

a) *q = *q - 1
b) **q

p[1] = p[1] - 1
= &a[1] - 1
= &a[0]

*q
= ~~*~~ ~~&~~ p[1]
= *p[1] = ~~*~~ ~~&~~ a[0] = a[0] = 10

```
void fun(int*);
void main(){
 int a[4]={10,20,30,40};
 fun(a);
 pf("%d %d",a[0],a[1]);
}
void fun(int *p){
 ++p;
 *p++;
 ++*p;
}
```

fun(a) → &a[0]

10 20

// &a[1]

⇒ *p++

| a[0] | a[1] | a[2] | a[3] |
|---|---|---|---|
| 10 | 20 | ~~30~~ 31 | 40 |
| 100 | 104 | 108 | 112 |

*(p++) Postincrement

~~&a[0]~~ ~~&a[1]~~ &a[2]

p

(i) *p

(ii) p = p+1 ✓

++(*p)

*p = *p+1

a[2] = a[2]+1

= 30+1

= 31

p = &a[2]

*p = *&a[2]

= a[2]

```
void main(){
    int a[]={10,20,30,40};
    f(a);  → &a[0]
    ==
}
```

```
void f(int *p)
{
    =
}
```

```
void f(int a[])
{
    =
}
```

internally int *a

```
void main(){
    int a[] = {10,20,30,40};
    int n;
    n = sizeof(a)/sizeof(int);
```

16 4 4

```
    f(a)
      =
    }
```

```
void f(int *a)
  {
    =
  }
```

```
void f(int a[])
  {
  int n;
  n = sizeof(a)/
        sizeof(int);
  }
```

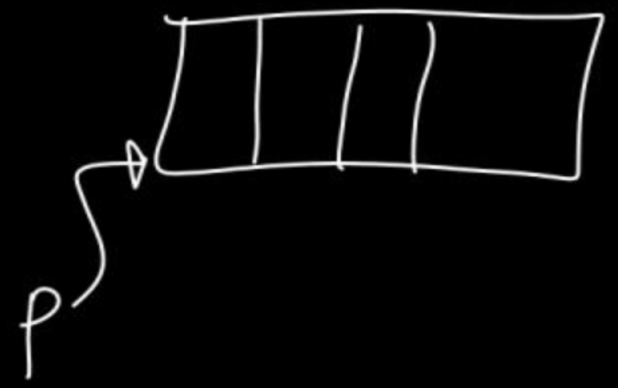
```
int f(int *);
void main(){

    int a[] = {10,20,30,40};
    int sum;
sum= f(a);

}
```

```
int f(int *p)
{

}
```



```
void main(){
    int a[3][3] = {{1,2,3},{4,5,6},{7,8,9}};

    fun(a);
        =
    }
```

$a \rightarrow \&a[0]$

Address of (an array of 3 integer)

Pointer to (an array of 3 integer)

```
void fun( int (*P)[3] )
{
    =
}
```

```
void fun( int P[ ][3] )
{
    }
```

internally int (*P)[3]

Pointer

int a[3][3] = {1,2,3,4,5,6,7,8,9};

a → &a[0] → Address of (An array of 3 integer)

a[0] → &a[0][0] ⇒ Pointer to (An array of 3 integer)

a[0][0]

&a → & whole array → int (*p)[3]

Address of (An array [2][3]) ← &a

int (*p)[2][3]

int *p

int a[2][3][2];

a → &a[0] Add of a 2-D array [3][2]

a[0] → &a[0][0] ⇒ Add of 1-D array [2]

a[0][0] → &a[0][0][0]

&a → add. of 3-D array

Pointer

int (*p)[3][2]

int (*p)[2]

int *p

int (*p)[2][3][2]

```
void main(){
    int a[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
    fun(a);
    pf("%d %d %d", a[1][1], a[1][2], a[2][0]);
    }
void fun( int (*p)[3]){
        ++p;
    (*p)[1] = (*p)[1] +1;
    }
```

6 6 7

$p = \&a[0]$

$++p$

$p = \&a[1] \Rightarrow *p \Rightarrow a[1]$

$(*p)[1]$

$a[1][1] = a[1][1] + 1$

Q

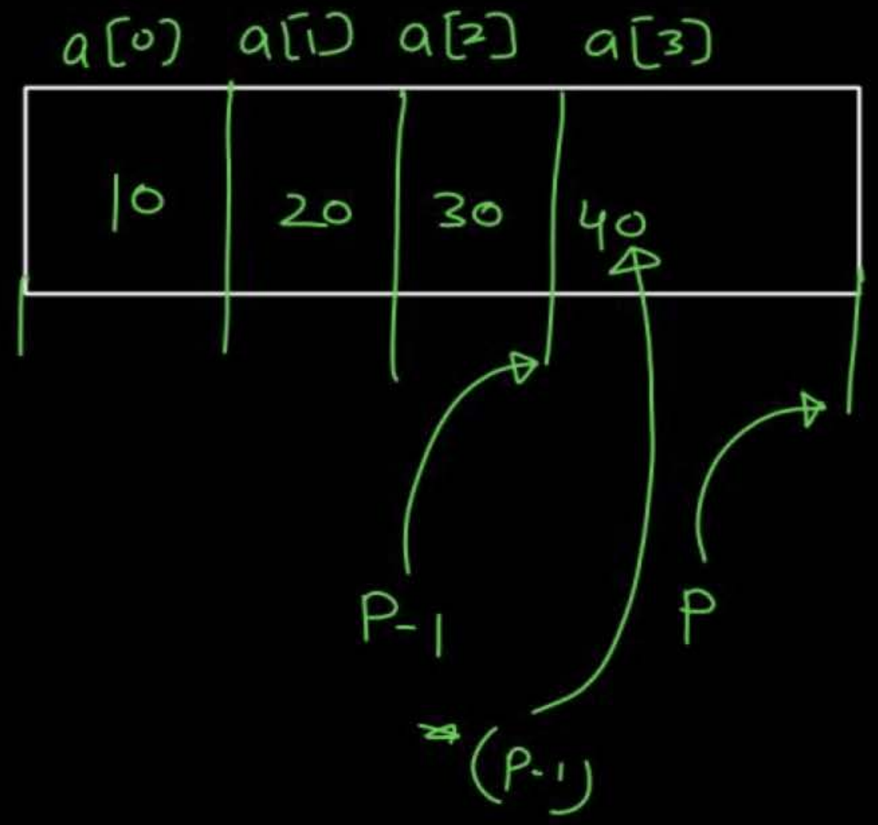
```
int a[4] = {10, 20, 30, 40};
int *p;
p = (int*)(&a+1);
pf("%d %d", *(a+1), *(p-1));
```

typecasting

$*(a+1) \rightarrow a[1]$

20

$*(p-1)$ → 40



Gate - 2010

A) -9
B) 5
C) 15
D) 19

```
int f(int *a, int n){
    if(n<=0) return 0;
    else if(*a % 2 == 0)
        return *a + f(a+1, n-1);
    else
        return *a - f(a+1, n-1);
    }

void main(){
    int a[] = {12, 7, 13, 4, 11, 6};
    pf("%d", f(a, 6));
    }
```

⇒ Even

⇒ odd

| a[0] | a[1] | a[2] | a[3] | a[4] | a[5] |
|---|---|---|---|---|---|
| 12 | 7 | 13 | 4 | 11 | 6 |
| 100 | 104 | 108 | 112 | 116 | 120 |

a, a, a, a

f(a 100, 6) 15

12 + f(a 104, 5) 3

7 − f(a 108, 4) 4

13 − f(a 112, 3) 9

4 + f(a 116, 2) 5

11 − f(a 120, 1) 6

6 + f(124, 0) 0

- Void Pointer
- Dynamic Memory allocation
- NULL Pointer
- Dangling Pointer
- Wild Pointer

05:00 PM

Scoping → last lecture
(Misc.)

*++*(++a)

$\boxed{a = a + 1}$

then use करो

*++*a

Preincrement

(i) First increase the value of variable

(ii) Then use it


THANK YOU GW
SOLDIERS !