unacademy

Arrays & Pointers - Part XI

Comprehensive Course on C- Programming

Pankaj Sharma • Lesson 27 • June 5, 2024

CS & IT Engineering

C Programming

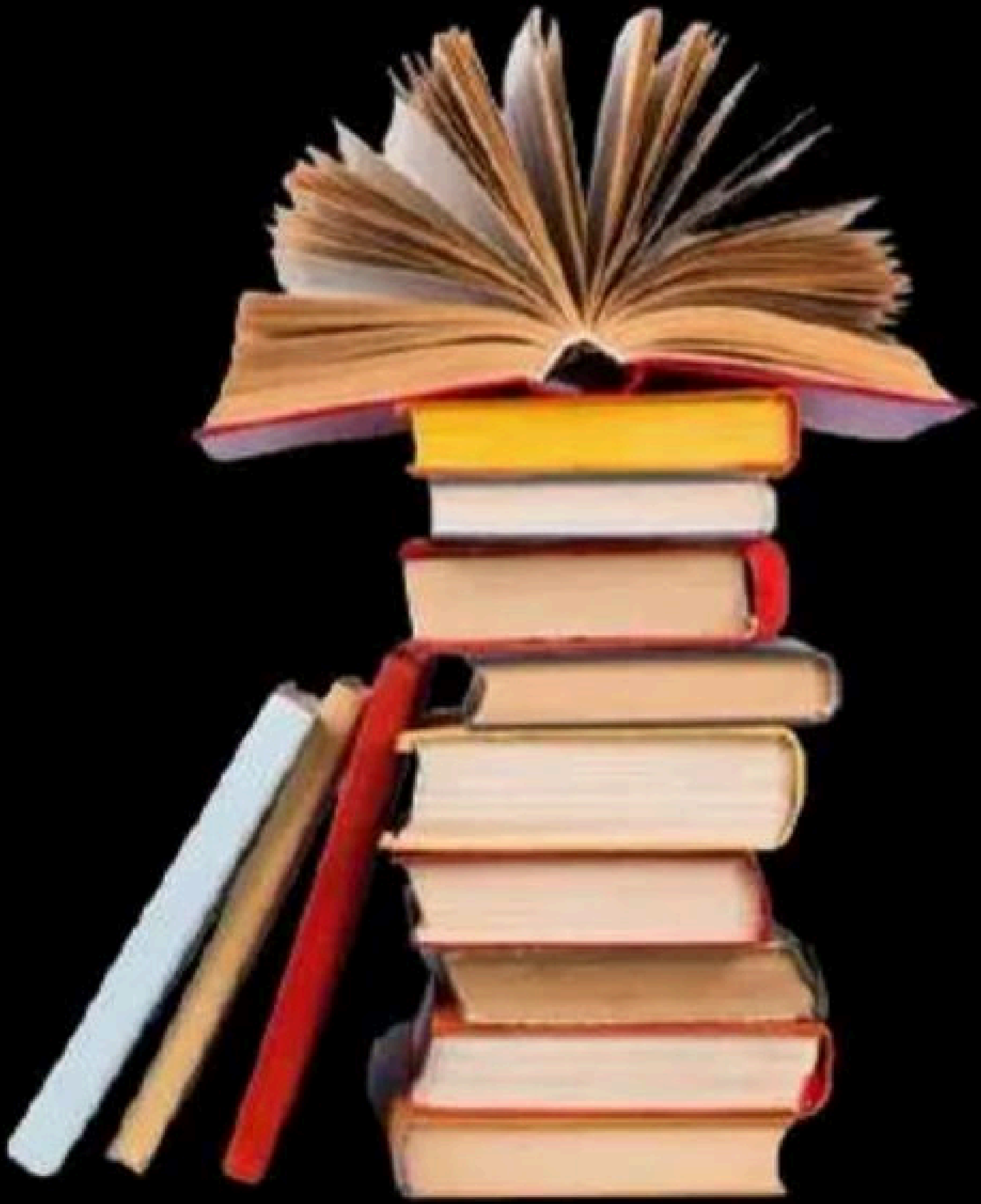
Arrays & Pointers-IX

Lecture Number- 27

By- Pankaj Sir

Topics

to be covered



1. Arrays & Pointers-IX

Q1. Which of the following are Invalid (MSQ)

✗ a) int b[ ][4]; Invalid Only decl.

✗ b) int b[ ]; 2nd Invalid Only dec.

✗ c) int b[2][ ][3] = {1,2,3,4}; Invalid Init.

d) int b[ ][2][3] = {10,20,30,40}; Init

1st dim

a, b, c

① Only declaration (without initialization)

[It is mandatory/compulsory to provide size of each dimension.]

② Initialization

There is flexibility only for 1st dim.

⇒ u can omit the size of 1st dimension

But this flexibility is only for 1st dim.

Q2:

P :

```
int arr[4] = {10,20,30,40};
printf("%d", *arr++);
```

→ Error ✗

Array-name++

Q :

```
int arr[4] = {10,20,30,40};
int *ptr = arr;
printf("%d", *ptr++);
```

✓

a) Only P [✓ b) Only Q] c) Both P and Q D) Neither P nor Q

Q3.

P :
```
int a[5) = {10,20,30};      ✓
printf("%d", 4[a]);         ✓
```

Q :
```
int 5[a) = {10,20,30};      Error
printf("%d", a[2]);
```

[✓ a)] Only P

b) Only Q

c) Both

D) Neither P nor Q

```
int a[5] = {10, 15, 20, 25, 30};
printf("%u", *(a+2) +6);
printf("%u", *(a + *(a+1) -12));
```

$\Rightarrow a[2] + 6$

$20 + 6 \Rightarrow 26$

$*(a + \underbrace{*(a+1)}_{a[1]} - 12)$

$*(a + 15 - 12) \Rightarrow *(a+3) \Rightarrow a[3]$

$\boxed{25}$

Q

$$\overset{a[3]}{\downarrow}$$

```
int a[5] = { 5, 3, 1, 2, 4 };
int *P[5] = { a, a+1, a+3, a+2, a+4 };
printf("%u %u", P[3][1], *(*(P+4)-2));
```

→ &a[2]

a] 5 3

b] 4 1

c] 2 1

D] 6 6

$\underbrace{P[3][1]}$

↓

$*(P[3]+1)$

$*(\&a[2]+1) \rightarrow \cancel{*}\cancel{\&}a[3] \Rightarrow a[3]$

Q

a[3]

&a[4]

```
int a[5] = { 5, 3, 1, 2, 4 };
int *P[5] = { a, a+1, a+3, a+2, a+4 };
printf("%u %u", P[3][1], *(*(P+4)-2));
```

a] 5 3

b] 4 1

[✓ c] 2 1]

D] 6 6

$*(P[4]-2)$

$*(\&a[4]-2)$

$\Rightarrow \cancel{*\&}a[2] \Rightarrow a[2] \Rightarrow \boxed{1}$

$$a[i][j] \Rightarrow *(a[i]+j)$$

$$p[3][1] \Rightarrow *(p[3]+1)$$

$$*(\&q[2]+1)$$

$$\cancel{*}\,\cancel{\&}q[3]$$

$$\Rightarrow q[3]$$

$$*\left(\underbrace{*(p+4)}_{\downarrow}-2\right)$$

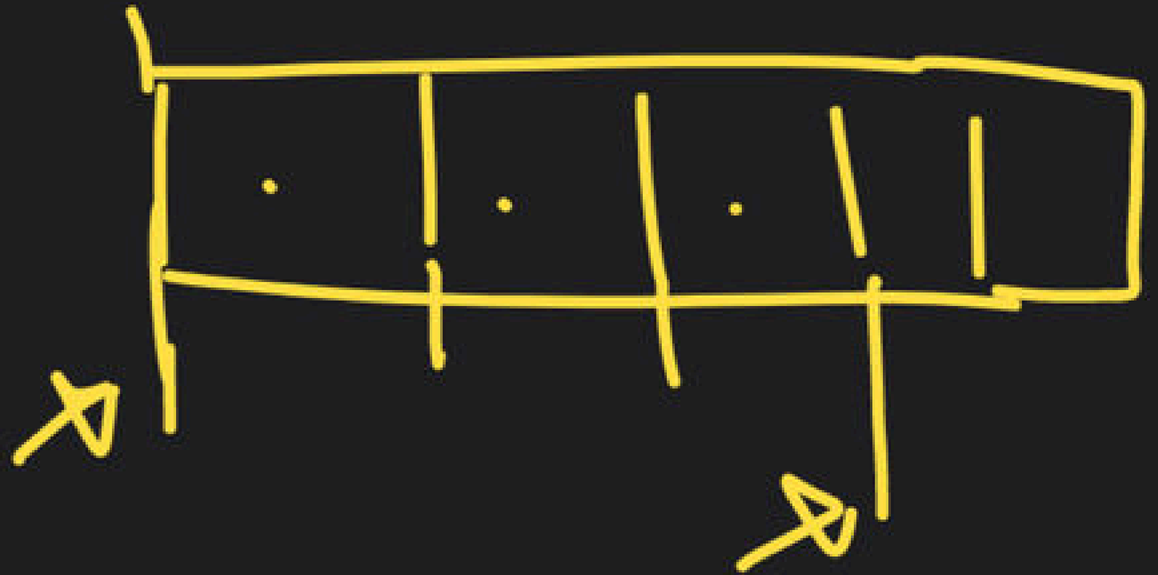
$$*\left(p[4]-2\right) \rightarrow \text{5th element of array p}$$

$$*\left(\&\underbrace{a[4]-2}\right)$$

$$\cancel{*}\cancel{\&}a[2] \Rightarrow a[2]$$

Q

```
int a[5] = {5,3,1,2,4};
int *p[5] = {a+3, a+1, a, a+2, a+4};
int **ptr = p+3;
pf("%u %u %u", ptr-p, *ptr-a, **ptr);
```



$\cancel{*\&}p[3]$

$*p[3]$

$\cancel{*\&}a[2]$

$a[2]$

①

3 2 1

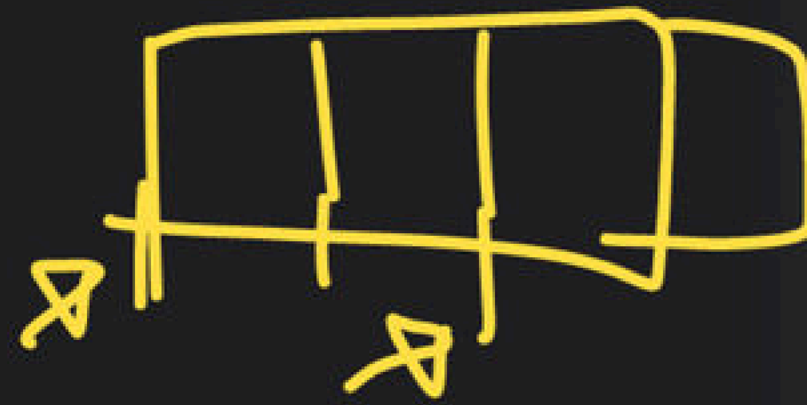
$\&p[3]-\&p[0]$

③

$p[3]-a$

$\&a[2]-\&a[0]$

②

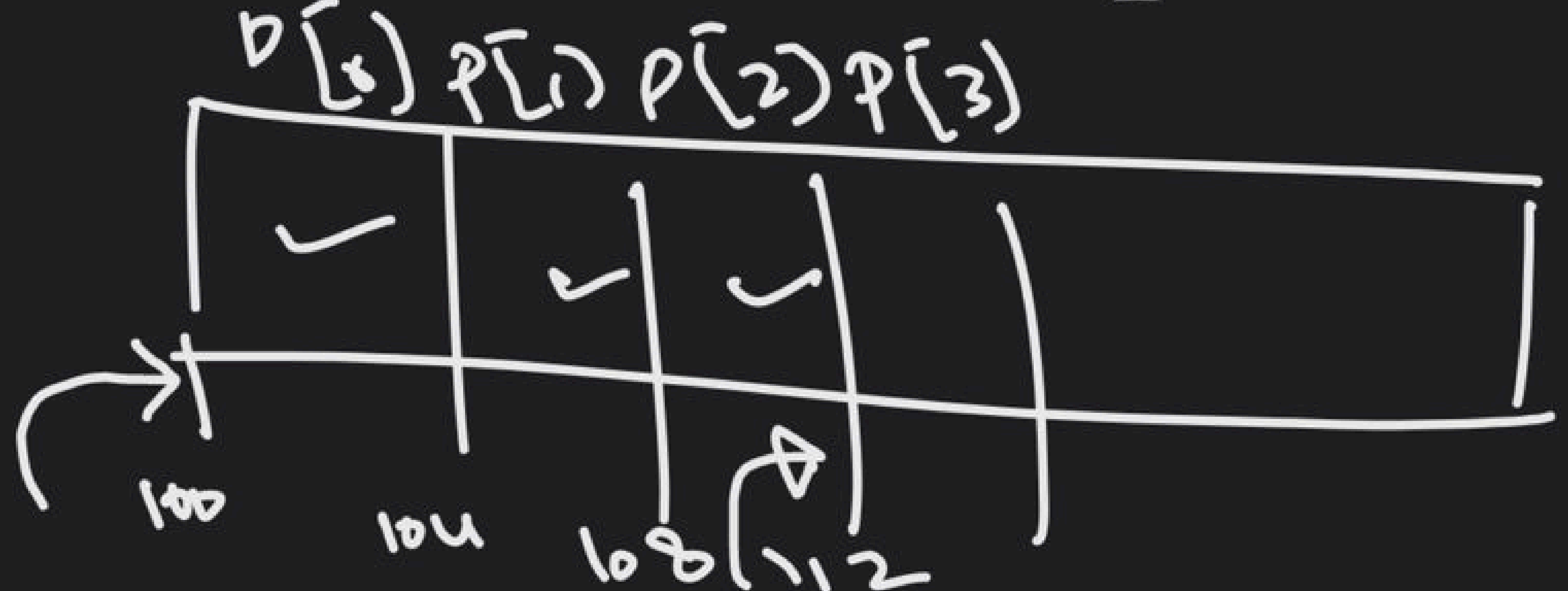


```
int a[5] = {5,3,1,2,4};
int *p[5] = {a+3, a+1, a , a+2, a+4};
int **ptr = p+3;        &p[0]+3 ⇒ &p[3]
pf("%u %u %u", ptr-p, *ptr-a, **ptr);
```

ptr - p

↓

$\&p[3] - \&p[0] \Rightarrow \frac{112-100}{4} = \frac{12}{4} = 3$



```
void func( int (*ptr)[2])
{
    **ptr+ = 1; →
     ptr++;
    **ptr* = 3;
}
void main(){
    int arr [2][2] = {0,1,2,3};
      func(arr); → &arr[0]
    pf("%d %d", a[0][0], a[0][1]); }
```

1 0	1	2	3

$**ptr = **ptr+1$

$arr[0][0] = arr[0][0]+1$

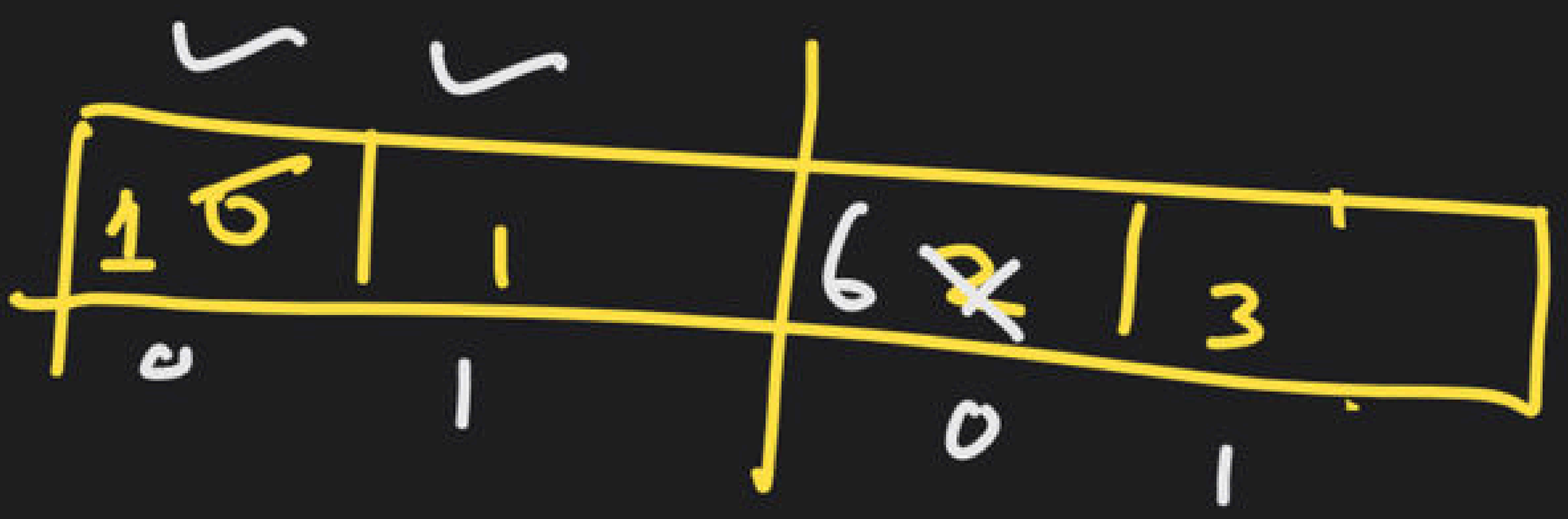
$ptr \quad \boxed{\&arr[0]}$

$*\cancel{*}\&arr[0]$

$*arr[0]$

$\cancel{*}\&arr[0][0]$

```
void func( int (*ptr)[2])
{
    **Ptr += 1;
    Ptr++;
    **Ptr* = 3;
}
void main(){
    int arr [2][2] = {0,1,2,3};
    func(arr);
    pf("%d %d", a[0][0], a[0][1]); }
```



$\Rightarrow$ Ptr = Ptr + 1

= &arr[0] + 1 = &arr[1]

func(arr) $\rightarrow$ &arr[0]

1 1

**Ptr = **Ptr * 3

$\cancel{*}$ $\cancel{*}$&arr[1] = **&arr[1] * 3

$\cancel{*}$&arr[1][0] = *&arr[1][0]*3

arr[1][0] = arr[1][0]*3

```
int a[3][2] = { 1, 3, 5, 7, 9, 11 };
int *ptr = a;
```

⇒ correct

⇒ Incorrect ✓

a

↓

&a[0]

Ptr

```
int a[3][2] = {1,3,5,7,9,11};
int *ptr = a[0];

ptr+=sizeof(int);
pf("%d", *ptr);
```

↓

5

Ptr = Ptr + 2

Assume size of int

↳ 2 byte

⇒ Run

↓

9

int a[3][2] = {1, 3, $\overset{6}{\cancel{5}}$, 7, 9, 11 };

int *ptr = a[1];

++*ptr++;

pf("%d", *ptr); ⑦

ptr

$\boxed{\&a[1][0]}$

a) 9

b) 5

c) 7

d) C.E

++(*(ptr++)) ①

(i) ++(*ptr)

(ii) ptr = ptr + 1

a) *ptr = *ptr+1

~~b)~~ *ptr (use)

Q

```
int a = 5, b = 10, c = 15;
int *p[3] = {&a, &b, &c};
pf("%d", *p[*p[1]-8]);
```

50 Questions
↓
{ data types & operators }

15

$*p[\cancel{*}\cancel{\&}b - 8]$

$*p[b-8] \Rightarrow *p[10-8)$

$\Rightarrow *p[2] \Rightarrow \cancel{*}\cancel{\&}c \Rightarrow c$

```
int a[] = {10,20,30,40,50};
int *p[] = {a, a+3, a+4, a+1, a+2};
int **ptr = p;
ptr++;
pf("%d %d", ptr-p, **ptr);
```

9th

ptr | &p~~[0]~~ &p[1]

$\&p[1] - \&p[0]$ → $\boxed{1}$

→ $\boxed{40}$

$*\cancel{*\&}p[1] = *p[1] = \cancel{*\&}a[3]$

$= a[3] \rightarrow 40$

unacademy

THANK YOU!

Here’s to a cracking journey ahead!