

# CS & IT ENGINEERING

Programming in C

Arrays and Pointer – 5

DPP 02 Discussion Notes

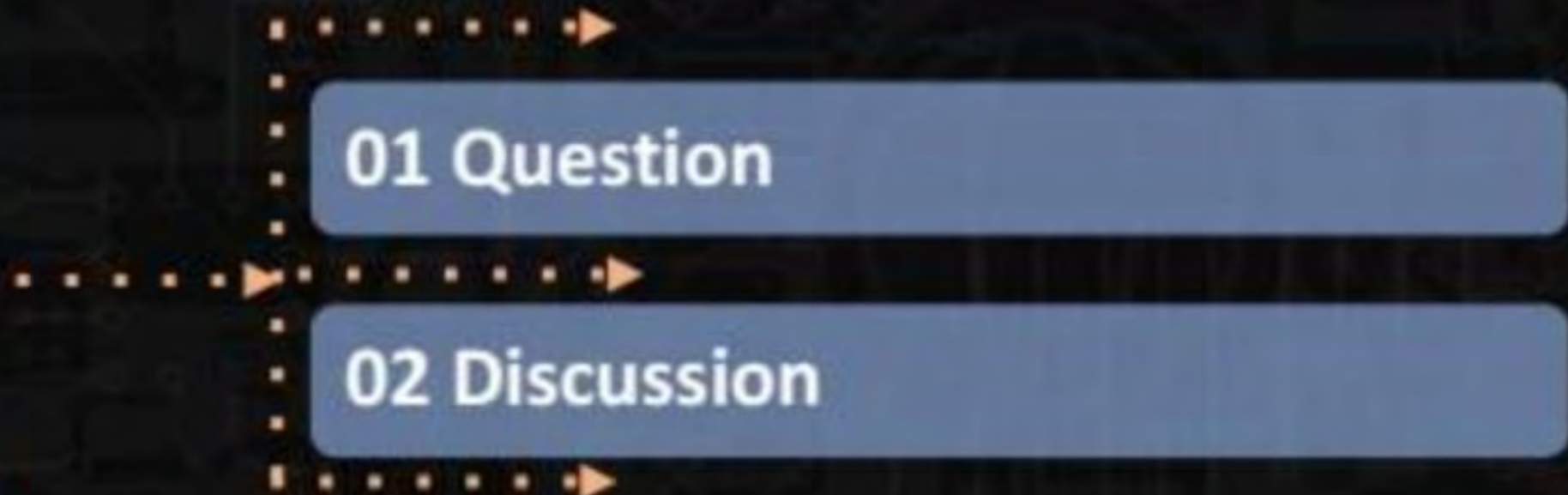


By- Pankaj Sharma sir





## TOPICS TO BE COVERED



01 Question

02 Discussion



**Q.1**

Consider the following program:

**[MCQ]**

#include&lt;stdio.h&gt;

int main()

{

int a[5]={5, 3, 1, 2, 4};

int \*p[5]={a, a+1, a+3, a+2, a+4};

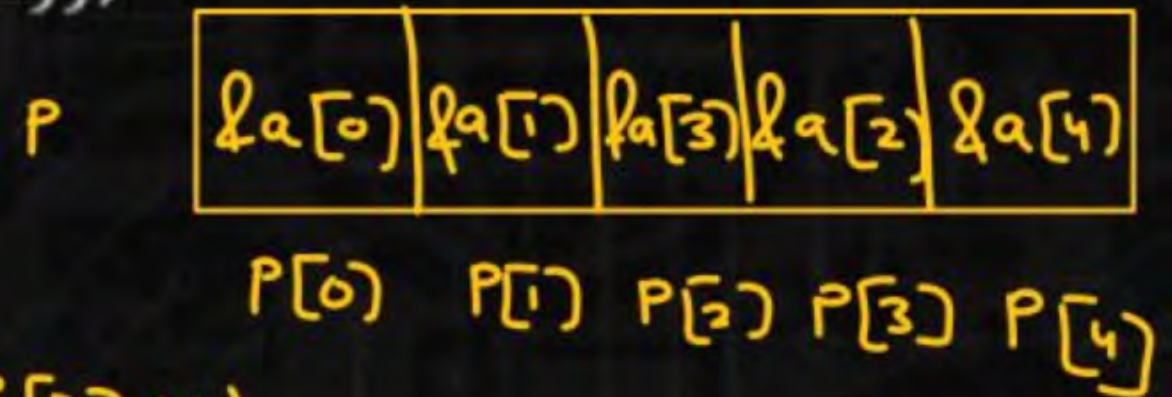
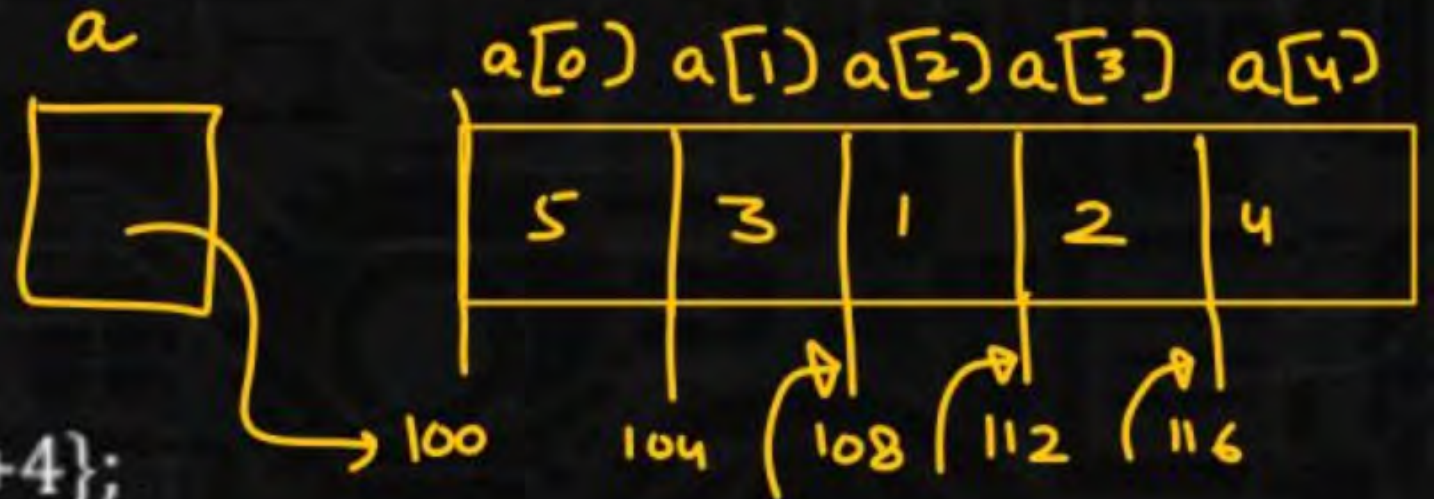
printf("%u\t%u", p[3][1], \*(\*(p+4)-2));

return 0;

}

The output is:

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



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

$$\Rightarrow *(\&a[2] + 1)$$

$$\&a[3] = a[3]$$

Garbage value

A.

5 3

B.

4 1

C.

2 1

D.

$$\begin{aligned} & \rightarrow (p[4] - 2) \\ & \rightarrow (\&a[4] - 2) \\ & \rightarrow \&a[2] \\ & = a[2] \end{aligned}$$



Q.2

Consider the following program:

```
#include <stdio.h>
int main()
{
    int a[]={2, 4, 6};
    int b[]={1, 3, 5};
    int *arr[]={a, b};
    printf("%u\t", *((arr+1)+2)); /*line 1*/
    printf("%u\t", **arr+3); /*line 2*/
    printf("%u", ***arr); /*line 3*/
    return 0;
}
```

Which of the following lines does not give ERROR?

A.

Line 1 only

☒ B.

Line 1 and Line 2 only

C.

Line 3 only

D.

Line 2 and Line 3

a

a[0]	a[1]	a[2]
2	4	6

b

b[0]	b[1]	b[2]
1	3	5

arr[0]	arr[1]
&a[0]	&b[0]

arr[0] arr[1]

[MCQ]



$*(arr+1)$

$\Rightarrow arr[1]$

$*(\&b[0]+2)$

$\Rightarrow 5$

$*(arr+3)$

$\&arr[0]+3$

$*arr[0]+3$

$\&a[0]+3$

$a[0]+3$

$***arr$

$\&arr[0]$

$arr[0]$

$\&a[0]$

$a[0]$

$arr[1]$

Invalid

arr

Error



**Q.3**

Consider the following function:

```
void f(int *p, int n)
```

```
{
```

```
  static int i;
```

```
  i=n-1;
```

```
  if(i<0) return;
```

```
  printf("%d\t", p[i]+p[n-i-1]);
```

```
  f(p,n-1);
```

```
}
```

If the array arr with elements [1, 2, 3, 4, 5] is passed as f(arr, 5), the output is-

~~A.~~

5 4 3 2 1

C.

6 5 4 3 2

~~B.~~

7 6 5 4 3

D.

2 3 4 5 6

arr

1	2	3	4	5
---	---	---	---	---

100

104

108

112

116

 $f(100, 5)$  $f(100, 4)$  $f(5)$  $f(100, 3)$  $f(4)$  $f(100, 2)$  $f(3)$  $f(100, 1)$  $f(2)$  $f(100, 0)$ ~~0 4 3 2 1~~ $P[0] + P[1+0-1]$  $P[0] + P[0]$  $f(6)$ **[MCQ]** $P[2] + P[3-2-1]$  $P[2] + P[0]$ 

3 + 1

= 4

 $P[1] + P[0]$ 

2 + 1



Q.4

Consider the following program:

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
int a[5]={5, 3, 1, 2, 4};
```

```
int *p[5]={a+3, a+1, a, a+2, a+4};
```

```
int **ptr=p+3;
```

```
printf("%u\t%u\t%u", ptr-p, *ptr-a, **ptr);
```

```
return 0;
```

```
}
```

The sum of the output is 321.

$a \Rightarrow \&a[0]$

$a+3 \Rightarrow \&a[0]+3$   
 $\&a[3]$

$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$
5	3	1	2	4
100	104	108	112	116

$\&a[3]$	$\&a[1]$	$\&a[0]$	$\&a[2]$	$\&a[4]$
$p[0]$	$p[1]$	$p[2]$	$p[3]$	$p[4]$
200	204	208	212	

$p$

$p = \&p[0]$

$ptr = \&p[0] + 3 = \&p[3]$

(ii)  $ptr - a$

$$\begin{aligned} & \&p[3] - \&a[0] \\ & \&a[2] - \&a[0] \\ & = \frac{108 - 100}{4} = \frac{8}{4} = 2 \end{aligned}$$

[NAT]



$ptr - p$

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

$212 - 200$

$$\frac{12}{4} = 3$$

(iii)

$ptr$

~~$\&p[3]$~~

$\Rightarrow *p[3]$

~~$\&a[2]$~~   
 $= a[2]$

$\&p[3]$   
 $ptr$



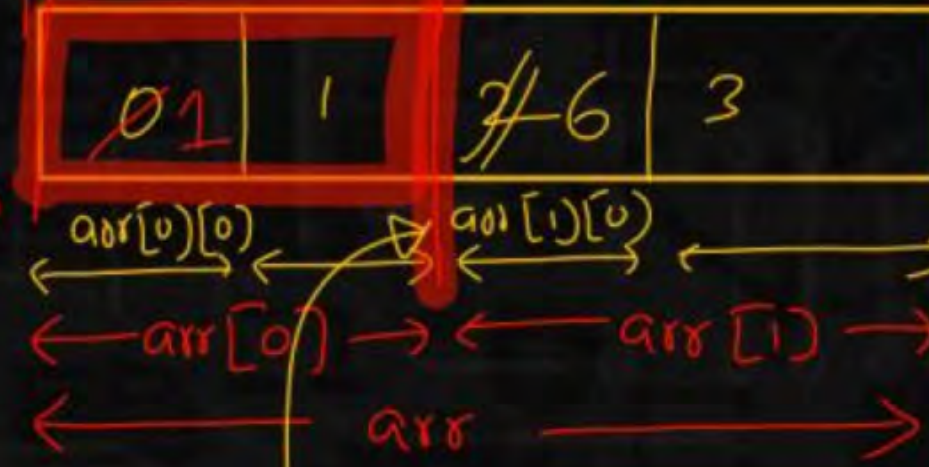
Q.5

Consider the following function:

```
void func(int (*ptr) [2])
```

```
{  
    **ptr+=1;  
    ✓ ptr++;  
    **ptr*=3;  
}
```

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



[MCQ]



$fun(arr)$   
 $\downarrow$   
 $\&arr[0]$   
 $**ptr \Rightarrow **\&arr[0]$   
 $\&arr[0]$   
 $\&arr[0][0]$

The array `arr[2][2]` with elements {0, 1, 2, 3} is passed to `func()`. What are the contents of the array after calling `func()`?

(ii)  $ptr++ \Rightarrow ptr$   $\&arr[1]$   
 $**ptr \Rightarrow **\&arr[1]$   
 $\&arr[1]$   
 $\&arr[1][0]$   
 $arr[1][0] = arr[0][0] \times 3$

A.

1 1 6 3

B.

0 1 2 3

C.

1 1 2 3

D.

Compilation Error.



Q.6

Consider the following program:

```
#include<stdio.h>
int main()
{
    int a[3][2]={1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23};
    printf("%u\t", a+1);
    printf("%u\t", *a+1);
    printf("%u\t", **a+1);
    printf("%u\t", ***a+1);
    printf("%u\t", &a+1);
    return 0;
}
```

Assume the base address of a is 100 and integer size is 2 bytes, the output is-

A.

124 112 102 2 106

C.

112 106 102 2 124

B.

124 102 112 5 106

D.

112 106 102 5 124

[MCQ]



$a[2][3][2]$

$*(*a)+1$

$\Rightarrow *a[0][0]+1$

$\Rightarrow \&a[0][0][0]+1 \times 2 = 102$

$\&a[0]+1 \Rightarrow \&a[0]+1 \times 12 = 112$

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

$*(*a)+1 \Rightarrow *a[0][0][0]+1 = 1+1 = 2$

$\&a+1 \times 24 = 124$

(C)



**Q.7**

Consider the following program:

```
#include<stdio.h>
```

```
int main()
```

```
{
```

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

```
    int *ptr=a;
```

```
    ptr+=sizeof(int);
```

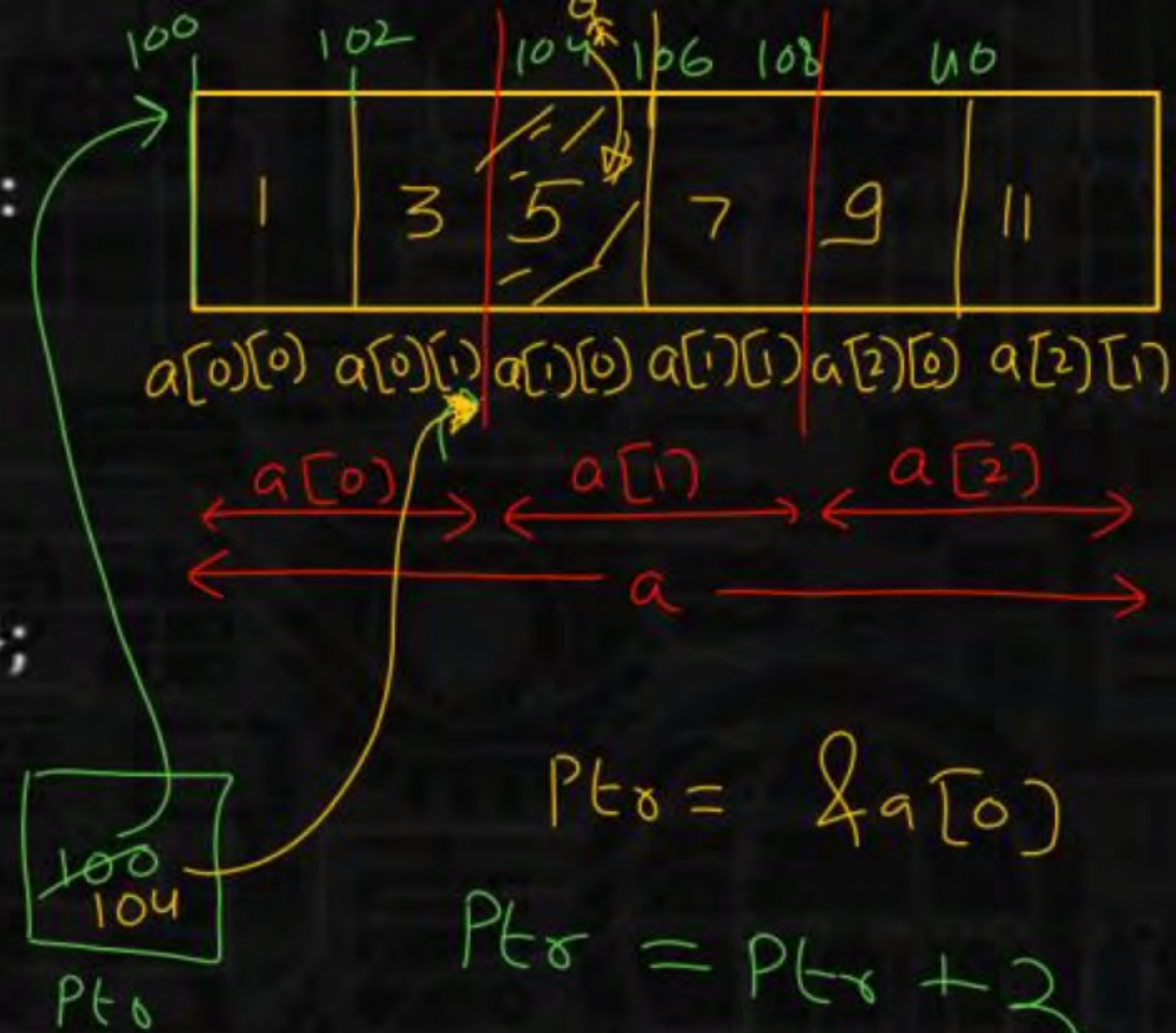
```
    printf("%d", *ptr);
```

```
    return 0;
```

```
}
```

(Assume size of int to be 2 bytes).

The output is 5.

**[NAT]**



**Q.8**

Consider the following program:

#include&lt;stdio.h&gt;

int main()

{

<sup>3</sup>  
int a[][2]={1, 3, 5, 7, 9, 11};

int \*ptr=a[1];

++\*ptr++;

printf("%d", \*ptr);

return 0;

}

The output is-

A.

5

B.

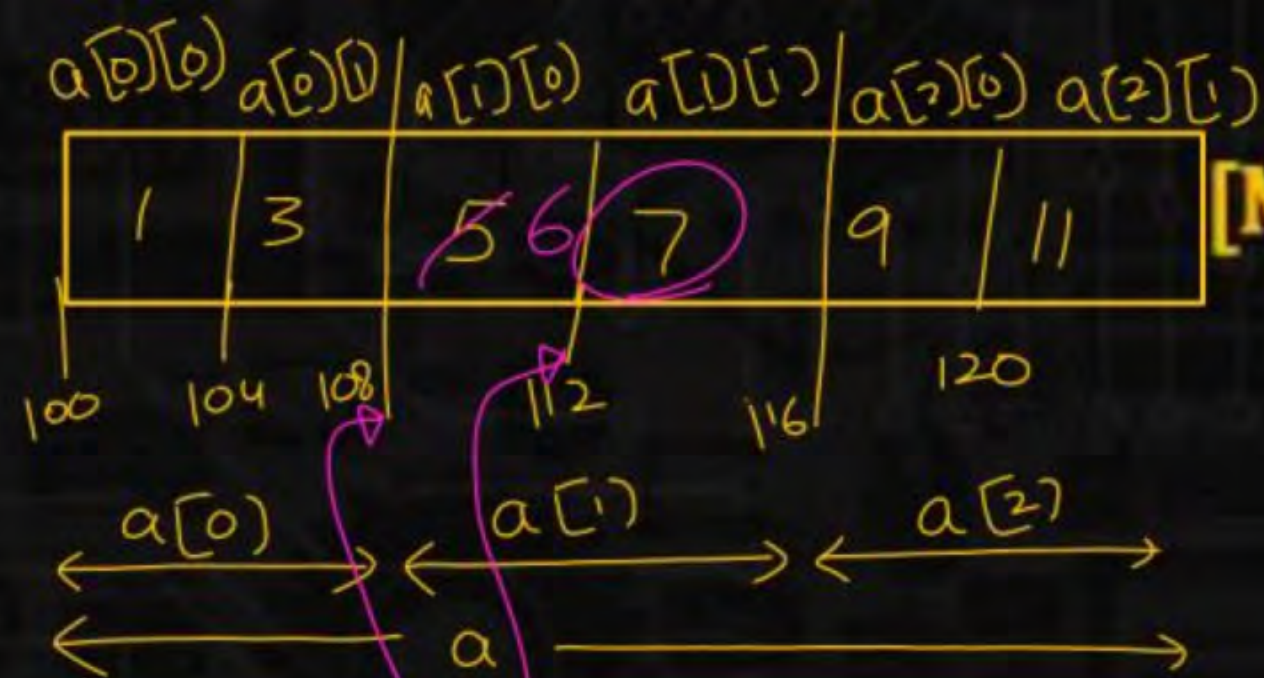
9

C.

7 ✓

D.

Compilation error

**[MCQ]**

Handwritten notes explaining the pointer arithmetic:

- $ptr = \&a[1][0]$
- $ptr = \&a[1][0] ++ (*ptr++)$
- $ptr = ptr + 1$
- $a[1][0] = a[1][0] + 1$
- $ptr = ptr + 1$



