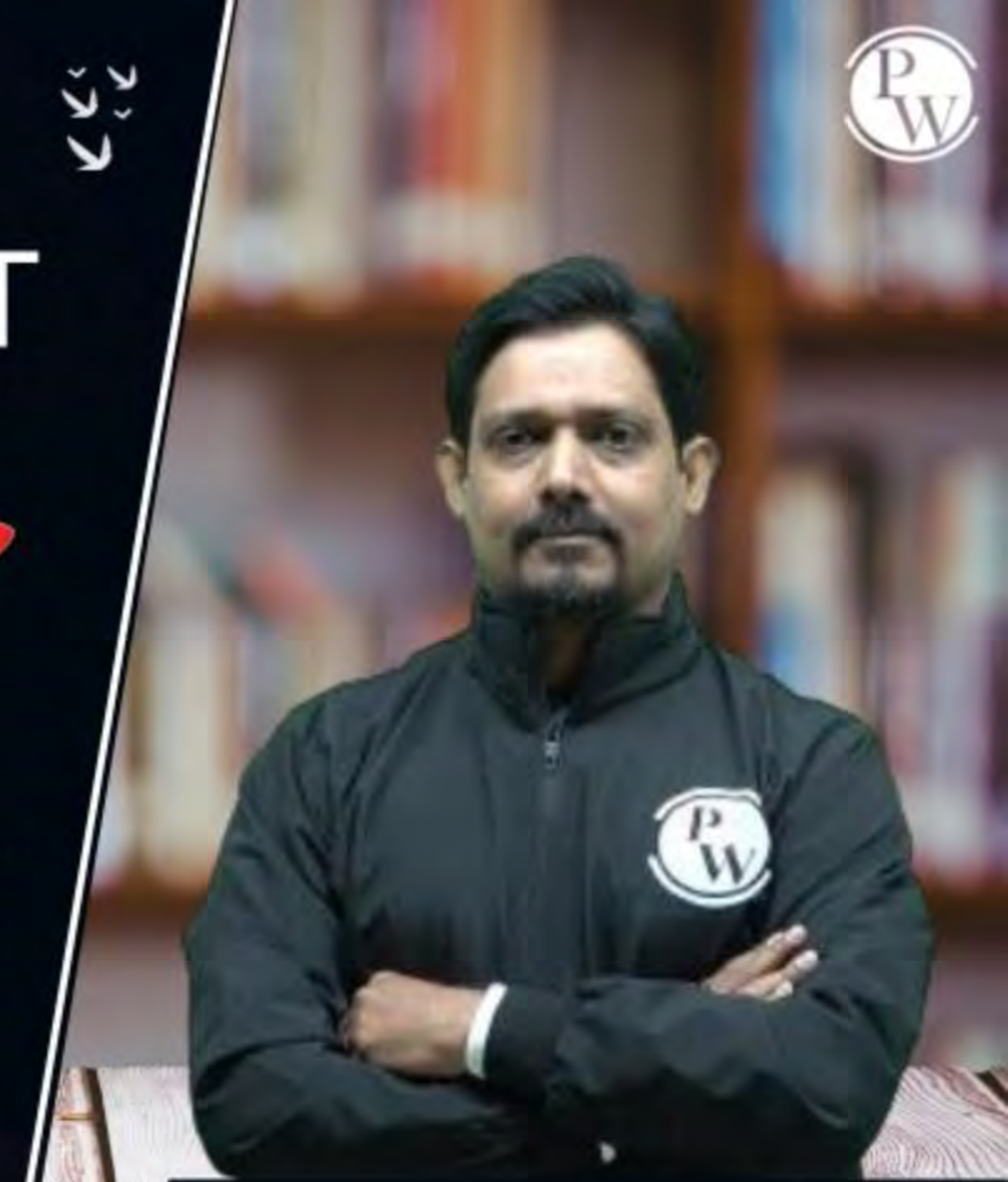
Computer Science & IT

C Programming



Array & Pointers

Lecture No. 04

By- Abhishek Sir

Recap of Previous Lecture

Topic 2-D array data type

Topic

Topic

Topic

Topic

Slide

Topics to be Covered

Topic practice problem array

Topic

Topic

Topic

Topic

Slide

2-D Array

```c
#include <stdio.h>

int main(){
   int a[2][2] = {{11,12},{13,14}};

   printf("\n %u",a);
   printf("\n %u",&a);
   printf("\n %u",*a);
   printf("\n %u",**a);

   printf("\n %u",a+1);
   printf("\n %u",*a+1);
   printf("\n %u",**a+1);
   printf("\n %u",&a+1);

   return 0;
}
```

Address of 1-D array

→ Address of 2D array

→ Address - of integer

→ integer

No value change

a - Add 1-D

* a Add - int

Calculate **a int

Slide

Question

What is the output of the following C program?

```
#include <stdio.h>
int main() {
      double a[2]={20.0, 25.0}, *p, *q;
      p = a;
      q = p + 1;
      printf("%d,%d", (int)(q - p), (int)(*q - *p));
      return 0;
}
```

(A) 4,8

(B) 1,5

(C) 8,5

(D) 1,8

100 108

p = 100

q = 108

1

typecast

$\frac{(108-100)}{8}$

$8/8 = \boxed{1}$

$(250 - 200)$

(int) 5.0

= 5

Slide

Question

Consider the following program

```
#include<stdio.h>
int main(){   100 104 108 112 116 120
    int arr[]= {10,20,30,40,50,60};
    int *ptr1= arr;
    int *ptr2 = arr + 5;
    printf("%d", (int)(ptr2 - ptr1));
    printf ("%d", (int)((char*)ptr2 - (char*) ptr1));
    return 0;
}
```

Output of the program is 520

$ptr_1 - \underline{100}$ integer

$ptr_2 - 100+5\times4 = \underline{120}$

$120 - 100 = \frac{20}{4} = \underline{5}$

$\left.\begin{matrix}120\\100\end{matrix}\right\}$ Address of character

$\frac{120-100}{1} = 20$

Size = 1 Byte

Slide

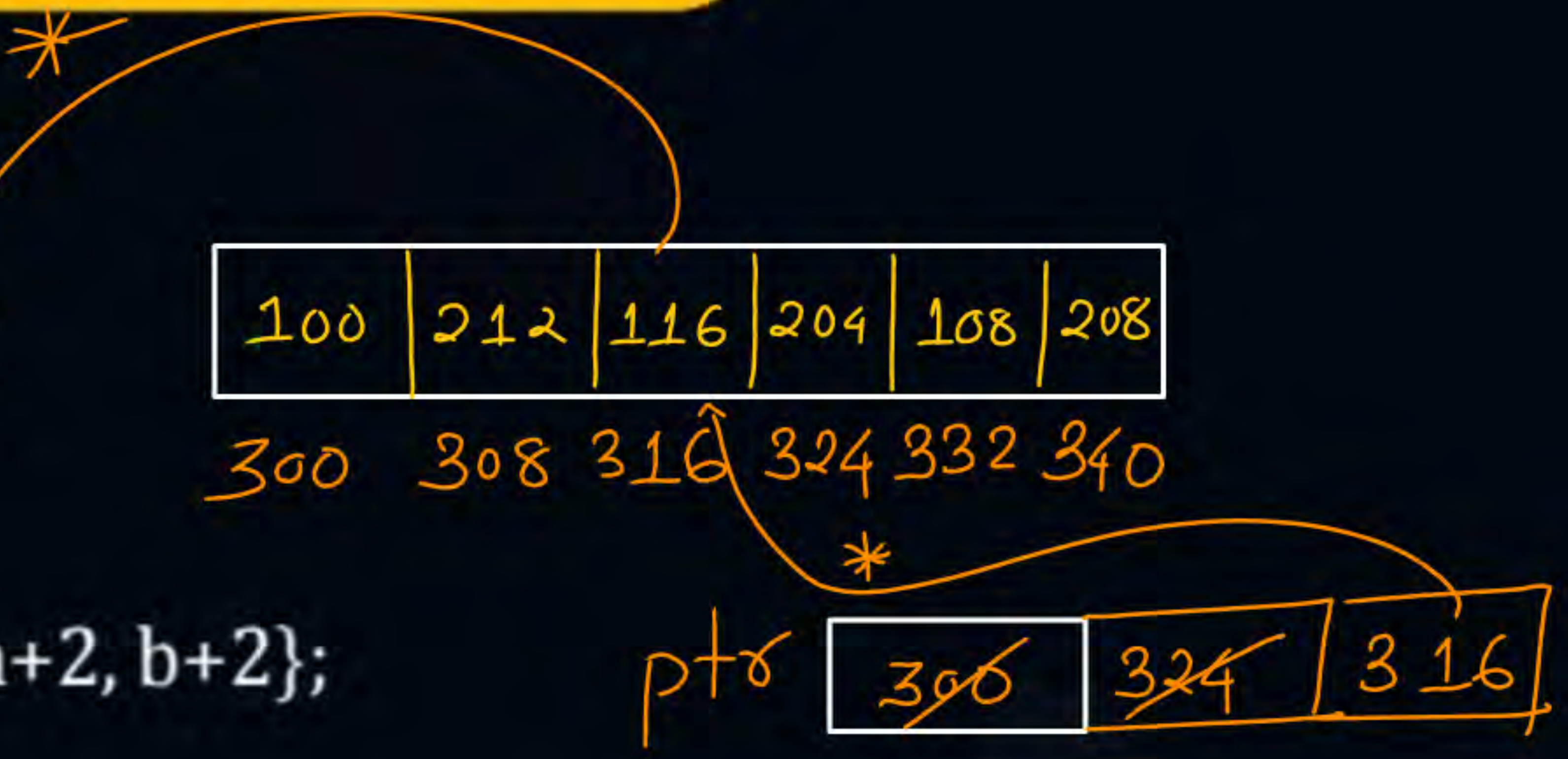
Question

```c
#include<stdio.h>
int main(){
    static int a[] = {14,27,73,40,50};
    static int b[] = {4,7,3,14,25};
    static int *p[] = {a, b+3,a+4,b+1,a+2, b+2};
    int**ptr=p;
    ptr+=3;
    printf ("%d", *ptr[-1]);
}
```

100 104 108 112 116

200 204 208 212 216

The output of the program is ___50___.



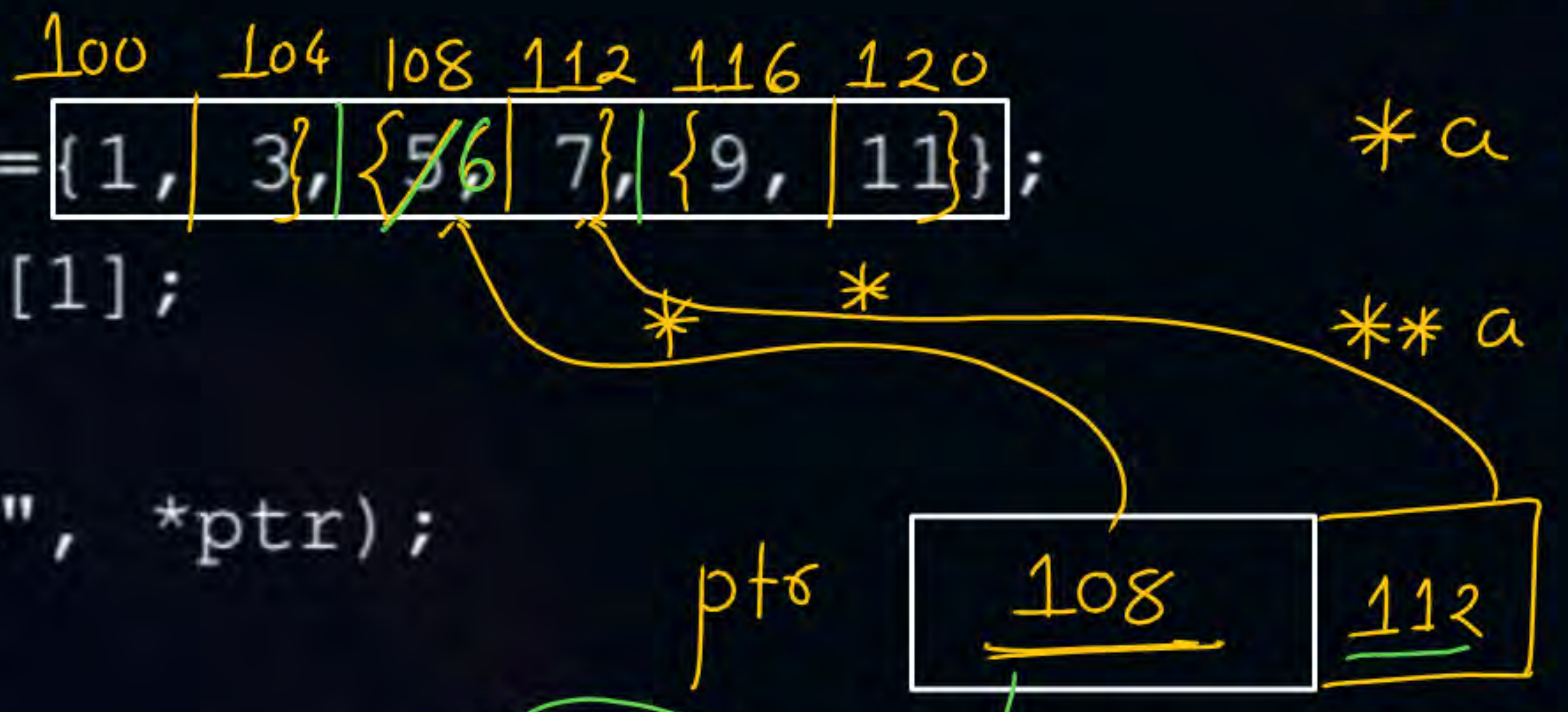
$$*\,ptr[-1] = *(*(ptr-1))$$

$$**(316)$$

Slide

Question

```c
#include<stdio.h>
int main(){
   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

a - Address 1D

*a Address of int

**a integer

++ *ptr++;

++ * ptr

++ * 108

$*(a+1) = *(100+1)$

$= *(100+1\times8)$

$= *(108)$

$= 108$

Question

Consider the following program

```
#include <stdio.h>
int main(){

    int a[][2] = {12,34,5,6,78,8,99};
    int *ptr= *(a+1);

    printf ("%d", *ptr++);
    ++*ptr;
    printf ("%d", *ptr);
    return 0;
}
```

The output of the program is 67

Size of array = 8B

100 104 108 112 116 120 124

7

5

7

ptr 108

112

$*(100+1)$

$*(108) = 108$

$*ptr = *(108)$

Slide

Question

What is the output of the following C program?

```
include <stdio.h>
int main() {
        double a[4]={20.0, 21.0,22.0, 23.0 }, *p, *q, *r;
        p = a;
        q = p + 2;
        r = p + 3;

        printf("%d", (int)(r - q)+ (int)(*r - *q)+(int)(*r - *p));
        return 0;
}
```

(A) 8

[✓ (B) 5]

(C) 2

(D) 1

100 108 116 124

p 100

q : 116

r : 124

$\frac{124 - 116}{8}$

(23.0 − 22.0)

23.0 − 20 0

1 + 1 + 3 : 5

Slide

Question

1 D- 4×4=16B

#Q What is the output of the following program ?

```
#include<stdio.h>
void main(){
static int a[]={62,61,19,40};
static int b[][4]={1,2,3,4,5,6,7,8,9};
int *ptr=a, sum=0;
int *ptr1 = b[1];
        for (int i =0;i< 4; i++, ptr++, ptr1++)
            sum+=*ptr-*ptr1;
    printf("%d", sum);
}
```

100 104 108 112

228

200 204 208 212 216 220 224 232

$*(200+1) = *(216)$

$=216$

ptr	100	104	108	112
ptr_1	216	220	224	228

i=0 = 0+62−5=57

i:1 = 57+61−6=112

i:2 112+19−7=124

i:3 124+40−8=156

Slide

GATE 2015

```
#include<stdio.h>
int main( ){
     static int a[] = {14,27,73,40,50};
     static int *p[] = {a, a+3,a+4,a+1,a+2};
     int**ptr=p;
     ptr++ ;
     printf ("%d%d", ptr-p, **ptr);
}
```

The output of the program is __________.

Slide

GATE 2015

What is the output of the following C code? (Assume that the address of X is 2000 (in decimal) and an integer requires four bytes of memory.)

```
int main() {
unsigned int x[4][3]={{1, 2,3},
                          {4,5, 6},
                      {7, 8, 9},
                      {10,11,12}};
 printf ("%u,%u,%u",x+3, *(x+3), *(x+2)+3);
     }
```

(A) 2036, 2036, 2036 ✓

(B) 2012, 4, 2204

(C) 2036, 10, 10

(D) 2012,4,6

3 element = 12

Address

without calculation

a - Address - 1-D

*a - Address - integer

**a - integer

X+3 = 2000+3 = 2000+3×12 = 2036

*(X+3) = *(2036) = 2036

*(x+2)+3

*(2000+2)+3 = *(2024)+3

= 2024+3 = 2024+3×4

= 2036

Slide

Question

Consider the following C program.

```
# include <stdio.h>
int main (){
int a[4][5] =  {{1, 2, 3, 4, 5},
                {6, 7, 8, 9, 10},
                {11, 12, 13, 14, 15},
                {16, 17, 18, 19, 20}};

printf("%d\n", *(*(a+**a+ 1)+4));
return (0);
}
```

The output of the program is ________.

	Col-0	Col-1	Col-2	Col-3	Col-4
Row-0	1	2	3	4	5
Row 1	6	7	8	9	10
Row 2	11	12	13	14	15
	16	17	18	19	20

2-D array 1-D array

$**a : a[0][0]$ $*a : a[0]$

$*(*(a+\underline{1}+1)+4)$

$*(*(a+2)+4) = a[2][4]$

Slide

Question

Consider the following C program.

```
# include <stdio.h>
int main (){
int a[4][5] =  {{1, 2, 3, 4, 5},
                {6, 7, 8, 9, 10},
                {11, 12, 13, 14, 15},
                {16, 17, 18, 19, 20}};

printf("%d\n", *(*(a+**a+ 1)+4));
return (0);
}
```

The output of the program is ________.

$5\times4 = 20B$

	Col-0	col-1	col-2	col-3	col-4
Row-0	1 100	2 104	3 108	4 112	5 116
Row 1	6 120	7 124	8 128	9 132	10 136
Row 2	11 140	12 144	13 148	14 152	15 156
	16 160	17 164	18 168	19 172	20 176

a - Add - 1D

*a - Add int

**a - integer

$*(*(a+2)+4)$

$*(*(100+2)+4)$

$*(*(140)+4)$

$*(140+4) = *(140+4\times4)$

$= *(156) = \underline{15}$

Slide

Question

What is printed by the following ANSI C program?

```
#include<stdio.h>
int main(int argc, char *argv[]){
    int x = 1, z[2] = {10, 11};
    int *p = NULL;
    p = &x;
    *p = 10;
    p = &z[1];
    *(&z[0] + 1) += 3;
    printf("%d, %d, %d\n", x, z[0], z[1]);
    return 0;
}
```

(A) 1, 10, 11

(B) 1, 10, 14 [D]

(C) 10, 14, 11

(D) ✓ 10, 10, 14

{10, ~~11~~ 14}

100 104

x ~~1~~ 10

200

p 104

*

$*(\&z[0]+1)+=3$

$*(100+1)+=3$

$*(104)+=3;$

10 10 14

Question

Consider the following ANSI C program

```
    int main() {
    int arr[5] [5];
    int i, j;
    for (i = 0; i< 5; i ++){
        for (j = 0; j < 5; j ++){
            arr[i] [j] = 23*i-j;
        }
    }
    printf ("%d", *(arr[0] + 13)+*(arr[1] + 13));
    return 0;
  }
```

What is the output of the above program?

(A) 208

[✓ (B) 109]

(C) 119

(D) 140

Question

Consider the following ANSI C program

```
int main() {
int arr[5] [5];
int i, j;
for (i = 0; i< 5; i ++){
     for (j = 0; j < 5; j ++){
          arr[i] [j] = 23*i-j;
     }
}
printf ("%d", *(arr[0] + 13)+*(arr[1] + 13));
return 0;
}
```

	col-0	col-1	col-2	col-3	col-4
Row 0	0	-1	-2	-3	-4
Row 1	23	22	21	20	19
Row 2	46	45	44	43	42
Row 3	69	68	67	66	65
Row 4	92	91	90	89	88

43

66 = 109

$*$ arr[0][13] + arr[1][13]

Question

Consider the following ANSI C program

```
int main() {
int arr[5] [5];
int i, j;
for (i = 0; i< 5; i ++){
    for (j = 0; j < 5; j ++){
        arr[i] [j] = 23*i-j;
    }
}
printf ("%d", *(arr[0] + 13)+*(arr[1] + 13));
return 0;
}
```

	col-0	col-1	col-2	col-3	col-4
Row 0	0	-1	-2	-3	-4
Row 1	23	22	21	20	19
Row 2	46	45	44	43	42
Row 3	69	68	67	66	65
Row 4	92	91	90	89	88

$a[0][5]\ *(*(100+0)+5) = *(120)$

$a[1][0] = *(*(100+1)+0)$

$*(*(100+1\times20)+0)$

$= *(120)$

GATE 2015

```
#include<stdio.h>
int main( ){
     static int a[] = {14,27,73,40,50};
     static int *p[] = {a, a+3,a+4,a+1,a+2};
     int**ptr=p;
     ptr++ ;
     printf ("%d%d", ptr-p, **ptr);
}
```

The output of the program is __________.

GATE 2015

```
#include<stdio.h>
int main( ){
      static int a[] = {14,27,73,40,50};
      static int *p[] = {a, a+3,a+4,a+1,a+2};
      int**ptr=p;
      ptr++ ;
      printf ("%d%d", ptr-p, **ptr);
}
```

The output of the program is __________.

Slide

2 mins Summary

Topic

practice problem

Topic

Topic

Topic

Topic

Slide

THANK - YOU