Computer Science & IT

C Programming

Array & Pointers

Lecture No. 03

By- Abhishek Sir

Recap of Previous Lecture

Topic — 1-D array

Topic — $a[i] = *(a+i) = *(i+a) = i[a]$

Topic — a Name of array address of first element

Topic

Topic

Slide

Topics to be Covered

Topic — Data type of 1-D array

Topic — Data type of 2-D array

Topic

Topic

Topic

Slide

1-D Array Declaration

int a[] = {

11,	12,	13,	14
100	104	108	112

};

Array Niwas – Same Address

2 Data types

Size

a - Address of integer : 100   4B

*a . integer

&a : Address of 1-D array   100   depends upon No of element in 1-D array

4X4B=16 B

&a[i] : Address of integer

& *(a+i)

Slide

1-D Array Declaration

int a[] = { 11, 12, 13, 14 };

11,	12,	13,	14
100	104	108	112

Array Niwas – Same Address

2 Data types

a - 100   a+1 = 104

*a .   *a+1 = 11+1 = 12

&a : 100   100+1 = 100+1 x Size of 1-D array

100+1x16 = 116

Slide

Difference Between two Address

ptr_1 and ptr_2 are Same panter

$$\boxed{ptr_1 - ptr_2 = \frac{ptr_1 - ptr_2}{\text{Sizeof data type}}}$$

$\underline{* ++x}$

Address increment

dereference

ptr_1 108

ptr_2 100

$ptr_1 - ptr_2 = 108 - 100 = 8$

Slide

Question

Out put of the program

```
#include <stdio.h>
void fun(int *x, int *y){
    *x++;
    *y = *y+10;

}
int main(){
    int a = 10, b=20;
    fun(&a, &b);
    printf("%d", a+b);
    return 0;
}
```

40

X $\boxed{10}$

Y $\boxed{30}$

X is an Address

* X ++ ,

*(100)

X $\boxed{100}\boxed{104}$

100 $\boxed{10}$

postfix higher precedence

++ will be done first

post increment Not in this expression

old value use hogi

Slide

Question

Out put of the program

```
#include <stdio.h>
void fun(int *x, int *y){
    (*x)++;
    *y = *y+10;

}
int main(){
    int a = 10, b=20;
    fun(&a, &b);
    printf("%d", a+b);
    return 0;
}
```

41

x $\boxed{11}$

y $\boxed{30}$

x $\boxed{100}\boxed{104}$

100 $\boxed{10}$

Slide

Question

3

Out put of the program

```
#include<stdio.h>
     int main(){
    int *i , b[] = {2, 3, 4, 5, 6};
     i =b;
     *++i;
     printf ("%d\t" ,*i); ✓
     i=i+3; ✓
      printf ("%d\t" ,*i); ✓
     }
```

100	104	108	112	116
2	3	4	5	6 ✓

(A) 2 3

(B) 3 4

✓ (C) 3 6

(D) Error

$i = \cancel{100}\ \boxed{104}$

$*{+}{+}i = *(104)$

$i = i+3 = 104+3 = 104+3\times 12$

$= \underline{116}$

Slide

Question

4,

#Q What is the output of the following program ?

```
#include<stdio.h>
int main(){
    int i , b[] = {2, 3, 4, 5, 6}, *p, **ptr ;
    p = b ;
    ptr = &p;
    p++;
    (*ptr)++;
    printf ("%d %d" ,*p,*(*ptr+2)) ;
}
```

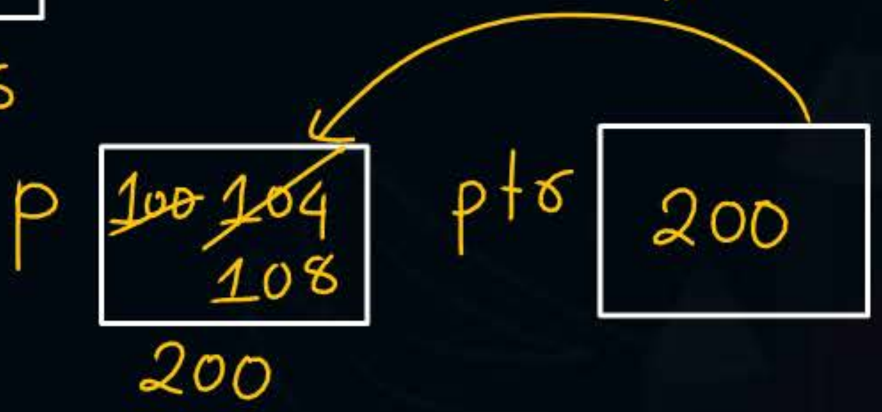
100 104 108 112 116

(A) 4 3

(B) 4 3

[✓ (C) 4 6]

(D) 3, 4



4 6

$*(*ptr+2)$

$*(108+2)$

$*(116)$

Slide

GATE 2015

panter 8B

ptr++,
++ptr,

```
#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);
}
```

Name of array
Address of first element

a

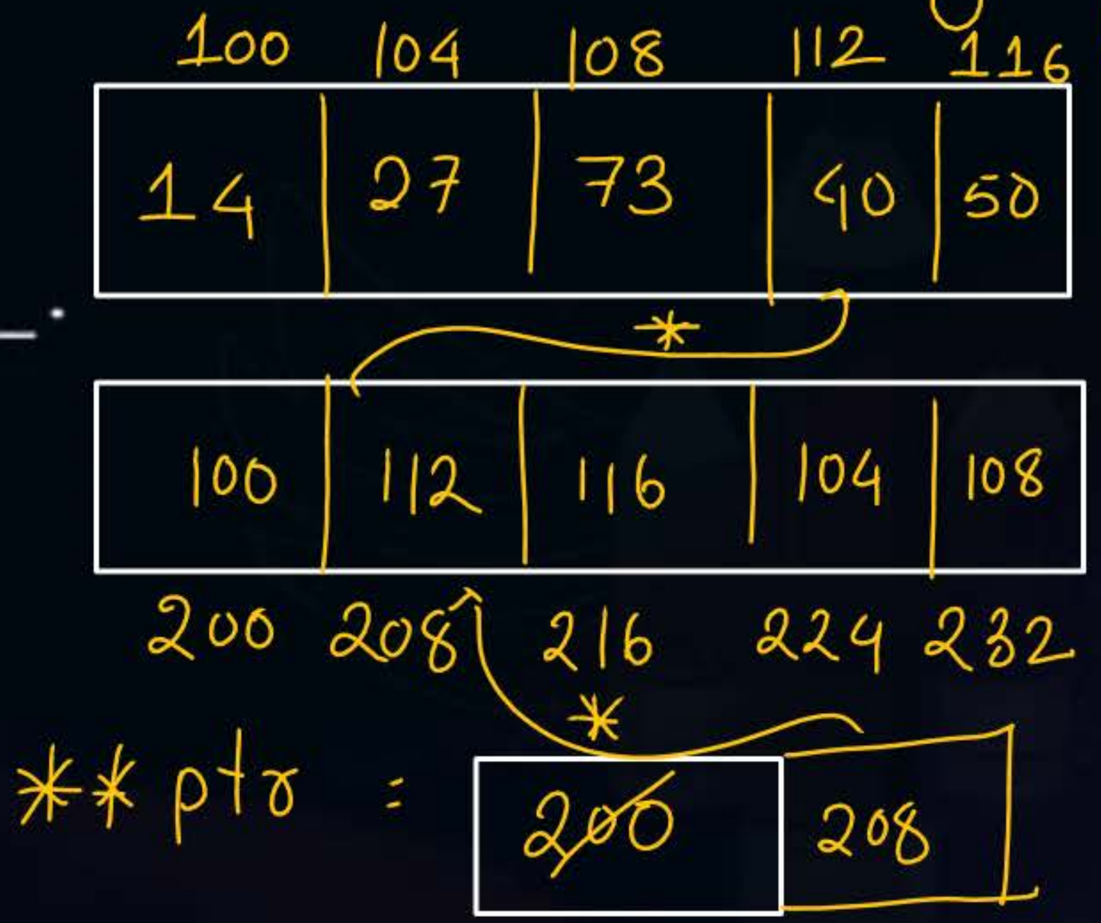
The output of the program is 140 ________.

*p - panter to integer

*p[] array of panter

$208 - 200$

$= 8/8 = 1$



Slide

2-D Array

$$\text{int } a[3][4] = \{\overbrace{\{1,2,3,4\}}^{\text{array}},$$

$$\{5,6,7,8\},$$

Array of 1-D array $\{9,10,11,12\}\};$

2-D array is array of 1-D array

Data type of 1^{st} element of 2D array

1-D array of integer

Slide

2-D Array

array of array

✓ [A] int a[2][2] = {1,2,3,4};

[B] int a[][2]; X

(C) int a[2][]; X

(D) int a[][]; X

(E) int a[][] = {1,2,3,4}; X

(F) int a[2][] = {1,2,3,4}; X

✓ (G) int a[][2] = {1,2,3,4,5};

warning ✓ (H) int a[2][2] = {1,2,3,4,5};

int a[2][3];

excess element

1	2	3	4	5	0

Slide

2-D Array

$$\boxed{a[i] = *(a+i)}$$

expand

$$a[i][j] = \boxed{*(a+i)}[j] = *(*(a+i)+j)$$

$**a$ is an integer

$a[0][0]$

$*(*(a+0)+0)$

$**a$

Slide

2-D Array

$$\underset{\substack{\uparrow \\ 2^{nd}\ \text{Raw}}}{a[2]}\underset{\substack{\uparrow \\ 3^{rd}\ \text{column}}}{[3]} = 12$$

	col-0	col-1	col-2	col-3
Raw-0	1	2	3	4
Raw. 1	5	6	7	8
Raw-2	9	10	11	[12]

$$*(*(a+2)+3)$$

Slide

2-D Array

1-D Size = 2×4 = 8Byte , 2-D = 4×4 = 16Byte

```
#include <stdio.h>

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

   printf("\n %u",a);      100
   printf("\n %u",&a);     100
   printf("\n %u",*a);     100
   printf("\n %u",**a);    11

   printf("\n %u",a+1);    100+1×8 = 108
   printf("\n %u",*a+1);   *100+1 . 100+1 = 100+1×4 = 104
   printf("\n %u",**a+1);  12
   printf("\n %u",&a+1);

   return 0;
}
```

{{11	12}	13	14}}
100	104	108	112

a : Address : 1-D array : 100

*a Address integer 100

value will not change

Data type will change

fitrat badal jah Hai

**a = integer

&a . Address of 2-D array   100+1 = 100+1×16 = 116

Slide

Array Niwas

&a Address of 2D

a - Address of '1-D

*a - Address of integer

value will not chage

**a integer

2-D Array

a - Addres 1-D ✓ 16B

*a Add- integer

	col-0	col-1	col-2	col-3
Row-0	100 1	104 2	108 3	112 4
Row. 1	116 5	120 6	124 7	128 8
Row-2	132 9	136 10	140 11	144 12

$a[2][3]$

$*(*(a+2)+3)$

$*(*(100+2)+3)$

$= *(*(100+2\times\underline{16})+3)$

$= *(*(132)+3)$

$= *(132+3) = *(132+3\times4)$

$= *(144) = \boxed{12}$

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

HW problem

practice Session

Slide

2 mins Summary

Topic 1-D array data type

Topic 2-D array data type

Topic a · Address 1-D

Topic *a · Address integer

Topic **a = a[0][0] integer

Slide

THANK - YOU