Computer Science & IT

C programming

Array & Pointers

Lecture No. 04

By- Abhishek Sir

Recap of Previous Lecture

- Topic
- Topic
- Topic
- Topic
- Topic

1-D array

Data type

Slide

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

2-D array

Data type

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;
}
```

typecast

2024

100 108

1

(5.0)

p = 100

q = 100 + 1 = 100 + 1×8 = 108

Double pointer

(108 - 100) = 108 - 100

= 8 8 = 8/8

(A) 4,8
(B) 1,5 ✓
(C) 8,5
(D) 1,8

Slide

2 D Array

Declaration

```
int a[2][2];
```

wrong

```
int a[][2];
int a[2][];
int a[][];
```

Initialization

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

correct Initialization

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

Slide

2 D Array

a[i][j]

Row major ordering

$= *(a+i)[j]$

$= *(*(a+i)+j)$

Two-Dimensional array access

a[i][j]

↑ Row ↑ column

Slide

Question

```c
#include <stdio.h>

int main(){
    int a[][3] = {11,12,13,14, 15, 16, 17 };

    for (int i=0;i<3;i++){
     for (int j=0;j<3;j++)
       printf("%d \t",a[i][j]);

     printf("\n");
    }
    return 0;
}
```

outer loop → 0

	0	1	2
0	11	12	13
1	14	15	16
2	17	0	0

a[0][1] a[0][2]

11 12 13

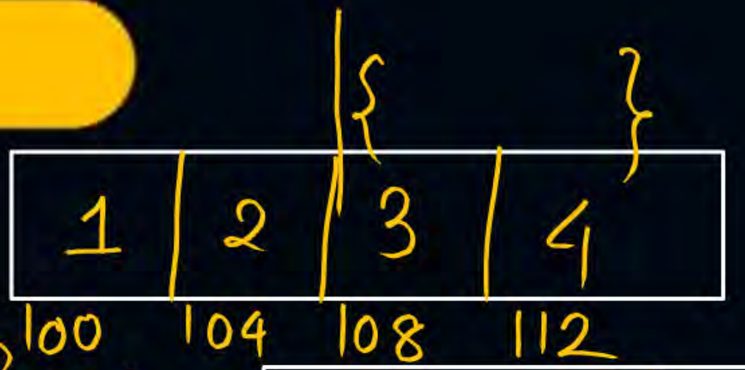
a[1][0] a[1][1] a[1][2]

14 15 16

Slide

2 D Array

$a[i][j] = \{\{1,2\},\{3,4\}\}$



a — Address of 1-D array ✓ — $a+1$

$a+1 \times \text{Sizeof} = 100+8 = 108$ (1-D array)

$*a$ — Address of integer ✓ — $*a+1 = 100+1\times4 = 104$

$**a$ — Integer ✓ — $**a+1 = 2$

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

Slide

2 D Array

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

$$a[i][j] = *(*(a+i)+j)$$

Slide

2-D Array

```
#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;
}
```

100+1 = 101

{ {11 | 12} {13 | 14} }

100 104 108 112

Data type Size

100 ✓
100 ✓
100 ✓
11

108
100+1 = 104

11+1 = 12

Address of 2D array; 100+1 = 100+1 × Sizeof data ty
100+1×16 = 116

Slide

2 D Array

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

a = 100 (Address - 1-D)

*a — 100 (Addes intger)

**a — 1

{1	2}	{3	4}	{5	6}
100	104	108	112	116	120

a+1 - 100+1×8 = 108

*a+1 - 100+1×4 = 104

**a+1 — 2

&a+1 100+1 = 100+1×24 = 124

Slide

2 D Array

double a[4][5] = { { 1.0, 2.0, 3.0, 4.0, 5.0 }, 100
{ 1.1, 2.1, 3.1, 4.1, 5.1 }, 108
{ 1.2, 2.2, 3.2, 4.2, 5.2 },
{ 1.3, 2.3, 3.3, 4.3, 5.3 } };

double = 8Byte

5 × 8 = 40

2D Size

20 × 8

160

a — 100 (Add) (1-D array)

*a 100 (Double Addres)

**a 1.0 (Double)

a+1 100+1 = 100+1×40 = 140

*a+1 100+1 = 100+1×8 = 108

**a+1 = 2.0

&a+1 2D array = 100+160 = 260

Slide

2 D Array

int a[4][4]

	0	1	2	3
0	100 1	104 2	108 3	112 4
1	116 5	120 6	124 7	128 8
2	132 9	136 10	140 11	144 12
3	148 13	152 14	156 15	160 16

a[3][2]

((a+3)+2)

1 → 2 → 3

((100+3)+2)

((100+3×16)+2)

((148)+2)

*(148+2)

*(148+2×4) = *(156) = 15

Integer Address

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

200

2000+3

2000+3 x Sizeof 1D

2000+3x12

2036

2036

*(2036)

2036

*(x+2)+3

*(2000+2)+3

*(2000+2x12)+3

*(2024)+3

Integer Add

2024+3

2024+3x4 = 2024+12

2036

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

Slide

GATE 2020

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 _____.

((a+1+1)+4)

((a+2)+4)

a[2][4]

1

15

Slide

GATE 2020

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 _____.

100 104 108 112 116
120 124 128 132 136
140 144 148 152 156

$*(*(a+1+1)+4)$

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

$100+2$

$\leq (*(100+2\times 20)+4)$

$*(*(140)+4)$

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

$*(156) = 19$

Slide

GATE 2020

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 _____.

100+2

100+2×20 = 140

*(140) + 3

140+3×4 = 140+12 = 152

If base Address is 100 and Size of element 4B

if a+2 is x (Addres)

if *(a+2)+3 is y

if *(a+2)-3 is z

~ *(140)-3

140-3×4 = 128

x+y+z is _____

140 + 152 + 128 = 420

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 _____.

HW

Memory diagram

Strong

2 mins Summary

Topic 2 D array

Topic 2 D array Data type

Topic practice

Topic

Topic

Slide

THANK - YOU