LECTURE - 35

ARRAYS (PART-04)

PROGRAMMING IN 'C'

```
int main() {
    int arr[] = { 100, 1000, 10000, 50000, 3000 };
    printf("%d", sizeof(arr));
    printf("%d", arr + 2);
    printf("%d", arr[0] + 5);
    printf("%d", arr[0] + arr[1]);
    return 0;
}
```

0 1 2 3 4

i = 0 to 4

arr = 5 elements (int)
= 5 × 4 = 20 Bytes ⇒ 20 Bytes

Assume &arr[0] = 500

arr = Address of first element

arr + 2 ⇒ 500 + 2 * sizeof(int)
⇒ 500 + 8 = 508

arr[0] + 5
⇓
100 + 5 = 105

arr[0] + arr[1]
100 + 1000
= 1100

Traversing an Array :

↳ Accessing each & every element of an array is called Array Traversing.

[loop] → Basically (safe) → for loop ✓

↳ (unsafe) → while loop ✓

```
int arr[4] = { 10, 12, 14, 16 };
              0   1   2   3

for (int i=0; i<4; i++) {
    printf("%d \n", arr[i]);
}
```

for : i = 0 → 4 { 0, 1, 2, 3 } ← 4 times

output

i = 0 → arr[i] → arr[0] ← 10

i = 1 → arr[i] → arr[1] ← 12

i = 2 → arr[i] → arr[2] ← 14

i = 3 → arr[i] → arr[3] ← 16

i = 4 ← false ← Exit

if no. of elements of an array is not defined

int arr[ ] = { a_1, a_2, a_3 ..... a_n }

Size of array = (no. of Elements * datatype of arr[0])

⇓

sizeof(array) = no. of Elements * size of (arr[0])

$$\text{no. of Elements } (n) = \frac{\text{size of (array)}}{\text{size of (array[0])}}$$ ← [One-Dimensional Array]

int arr[ ] = { 15, 16, 24, 30, 29, 18, 3, 4, 7 } ← 9 elements

(indices: 0, 1, 2, 3, 4, 5, 6, 7, 8)

```
int n = sizeof(arr) / sizeof(arr[0])

for (int i = 0; i < n; i++){
        printf("%d", arr[i]);
}
```

arrays01.c | arrays2.c | array3.c | array4.c | array05.c | arrayTraversing.c

arrayTraversing.c > main()

```c
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main(){
    int arr[] = {10,32,20,51,61,6,16,51,65,1,6,10,3,21,
    132,3,20,32,32,24};
    printf("the size of array is: %d\n", sizeof(arr));
    printf("the size of first element of array is: %d
    \n", sizeof(arr[0]));
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("The number of elements in arr are: %d\n", n);
    for (int i = 0; i<n; i++){
        printf("%d ", arr[i]);
    }
    return 0;
}
```

Code

```
the size of array is: 80
the size of first element of array is: 4
The number of elements in arr are: 20
10 32 20 51 61 6 16 51 65 1 6 10 3 21 132 3 20
32 32 24
PS>
```

Write a program to print all the even numbers from an integer array.

→ num % 2 == 0 ← Even

```
int main( ){
    int arr[ ] = { 1, 3, 4, 5, 10, 16, 12 };
    for ( int i=0 ; i< 7 ; i++ ) {
        if (arr[i] % 2==0) {
            printf("%d ", arr[i]);
        }
    }

    return 0;
}
```

0 1 2 3 4 5 6

7 → Sizeof (arr) / Size of (arr[0])

Dry Run

for loop → i = 0 → 7 { 0, 1, 2, 3, 4, 5, 6 }

i=0,	arr[0] = 1,	1 % 2 == 0 (false)		
i= 1,	arr[1] = 3,	3 % 2 == 0 (false)		
i= 2,	arr[2] = 4,	4 % 2 == 0 (True) →	print →	4
i= 3,	arr[3] = 5,	5 % 2 == 0 (false)		
i= 4,	arr[4] = 10,	10 % 2 == 0 (True) →	print →	10
i= 5,	arr[5] = 16,	16 % 2 == 0 (True) →	print →	16
i= 6,	arr[6] = 12,	12 % 2 == 0 (True) →	print →	12

Output

4
10
16
12

arrays2.c | array3.c | array4.c | array05.c | arrayTraversing.c | oddArr.c

oddArr.c > ...

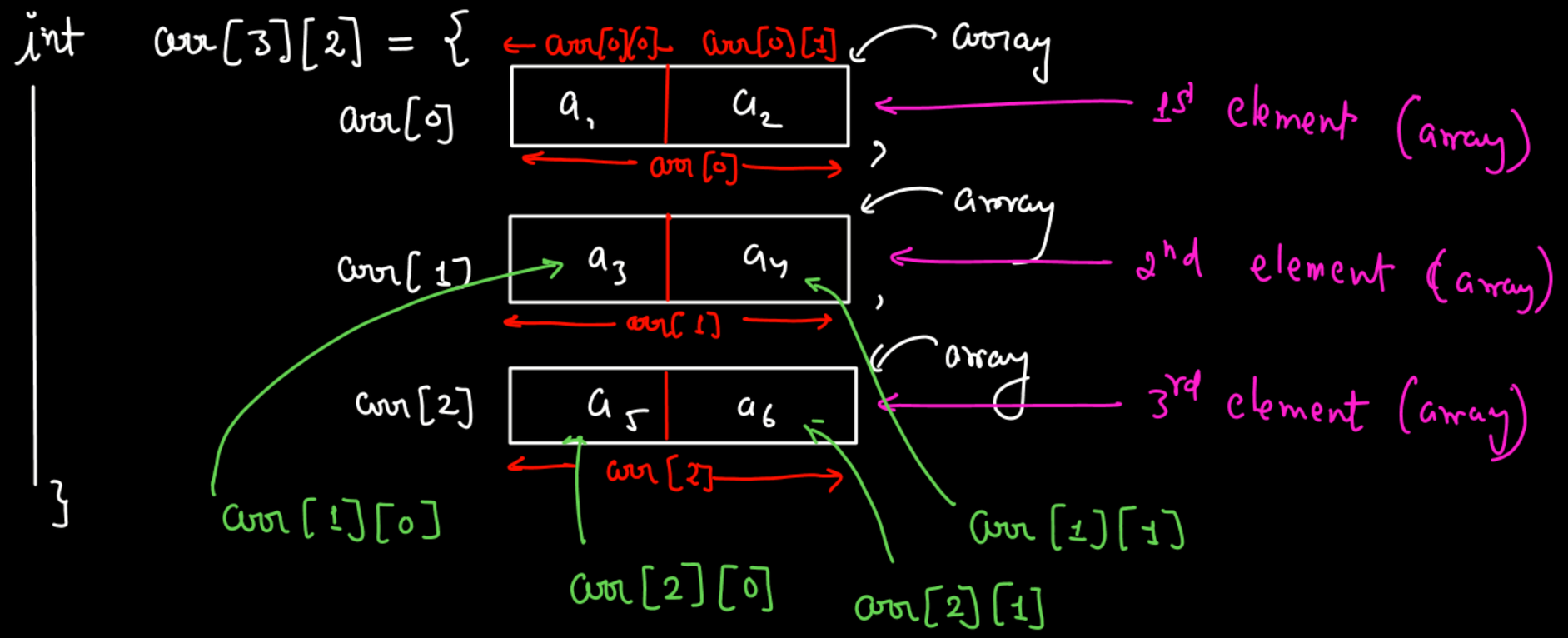
```c
// Printing the odd numbers from an array
#include<stdio.h>

int main(){
    int arr[] = {10,12,13,15,19,17,24,29,27,39,36,35,45,
    55,60};
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("List of Odd numbers:");
    for (int i = 0; i<n; i++){
        if(arr[i]%2 != 0){
            printf("%d ", arr[i]);
        }
    }
    printf("\nList of even numbers:");
    //printing the even number of an array
    for(int i= 0; i<n;i++){
        if(arr[i]%2==0){
            printf("%d ", arr[i]);
        }
    }
    return 0;
}
```

```
List of Odd numbers:13 15 19 17 29 27 39 35
45 55
List of even numbers:10 12 24 36 60
PS>
```

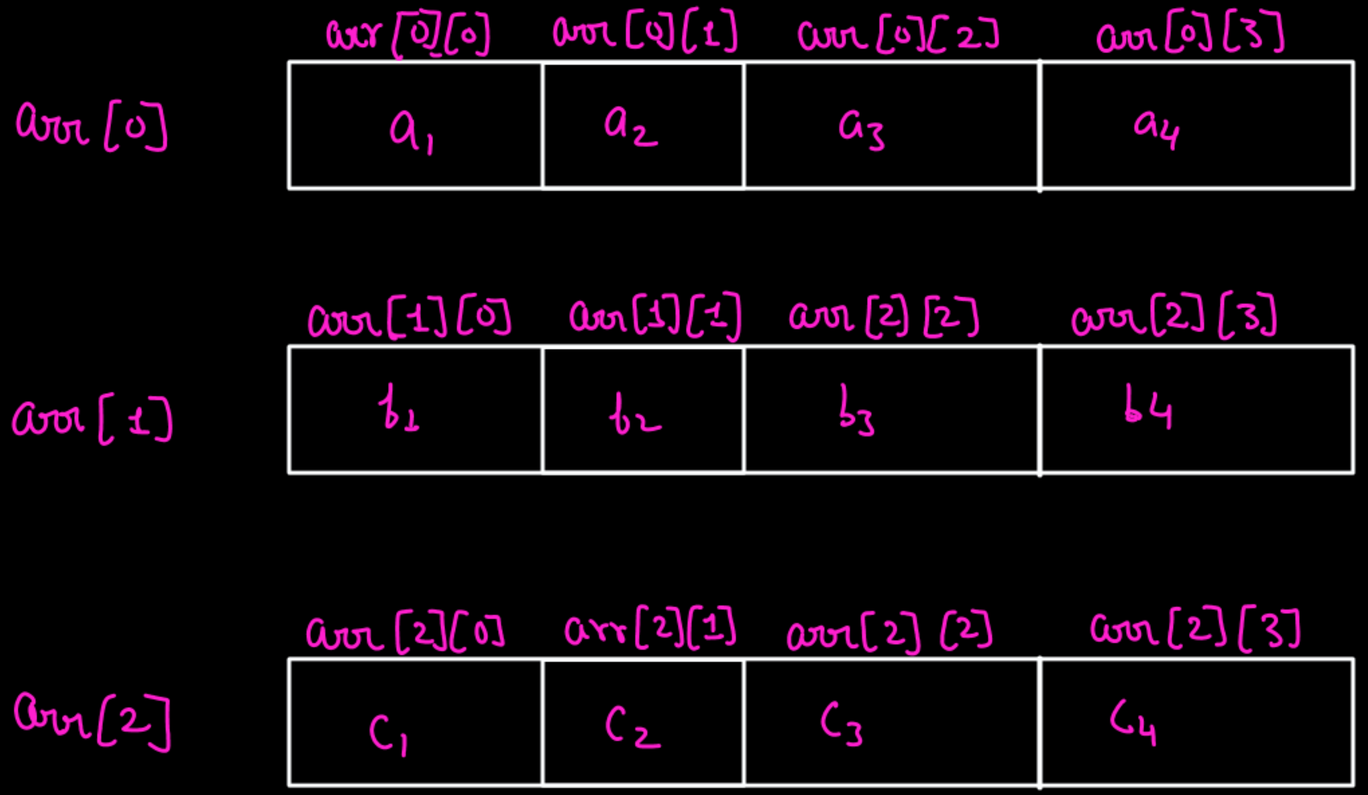
Two Dimensional Array :

↳ Array inside an array.

⇒ An element of an array consists an array.



int arr[3][4] = {



}

* 2D array act as a matrix

int arr[3] ← 3 elements

arr[a][b]

↓

'Array of 'a' arrays where each 'a' consist 'b' elements'

arr[3][5]

↓

'array of 3 arrays where each array consists 5 elements'