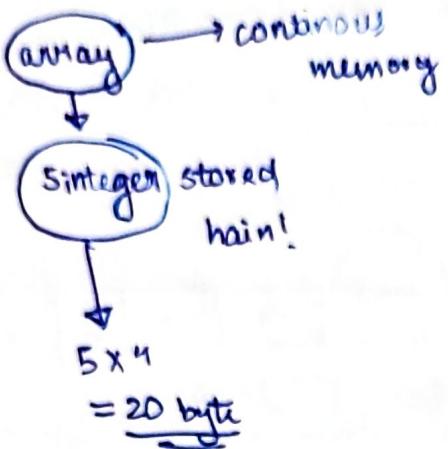


Week 3 Class 1 :-

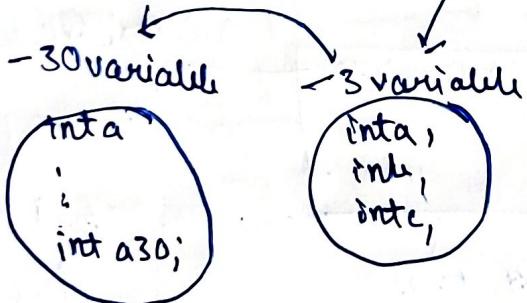
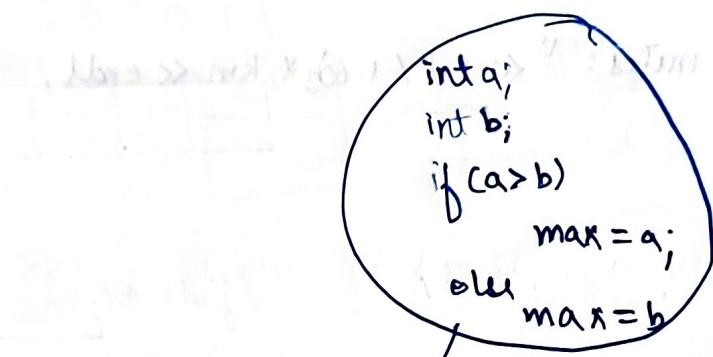
Array :- DS hai

- similar type of data store kota hai



Why array is needed?

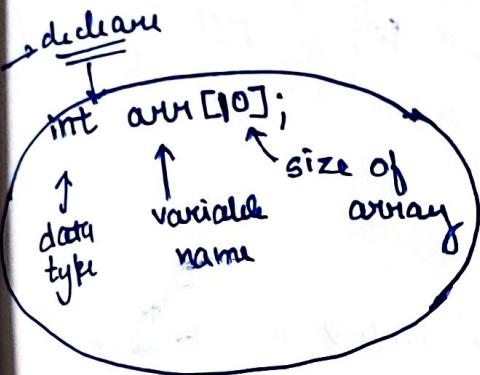
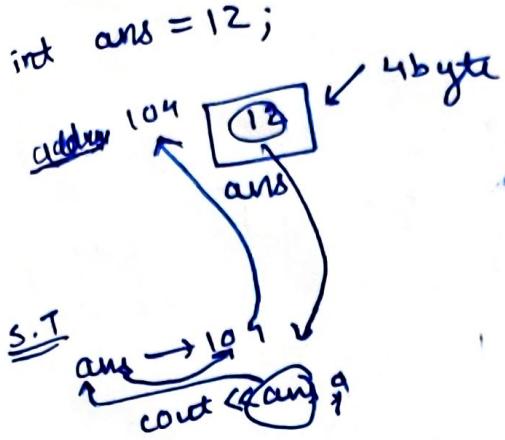
2 variables



- 30,000 variables k liye ?

int arr[30000];

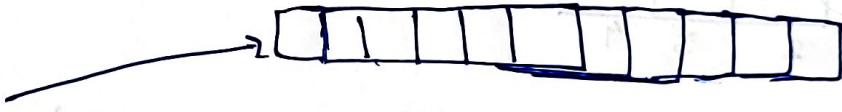
30,000 memory locations allocated to it.
30,000 variables created



`int arr [10];`

↓
10 integer ka

continuous space assign hojata hai
memory

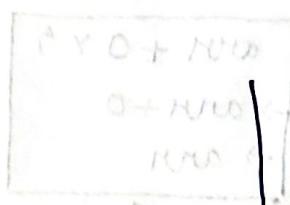


`int main () {`

`int arr[7];`

`cout << "Array created" << endl;`

`return 0;`



Array created

Declaring

`int main () {`

① `int arr[53];`

② `char a[106];`

③ `bool a[23];`

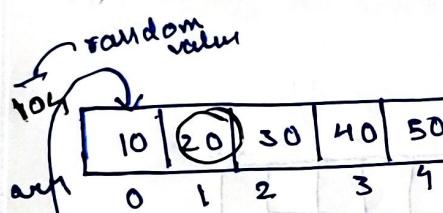
`int arr[] = { -,-,-,-,- };`

`int arr[5] = { -,-,-,-,- };`

Index and access in Array:-

10	20	30	40	50
0	1	2	3	4th index

5 size $[0 \rightarrow 4]$



`int arr[] = { 10, 20, 30, 40, 50 }`

`arr[0] = 10`

`arr[1] = 20`

`arr[2] = 30`

`arr[3] = 40`

`arr[4] = 50`

$\rightarrow arr + 0 \times 4$
 $\rightarrow arr + 0$
 $\rightarrow arr$

↑
value at this address

`arr[1] → value at (arr + 1 × 4)`

$\rightarrow 104 + 4$

$\rightarrow (108)$

`arr[3] → value at (arr + 3 × 4)`

$\rightarrow 104 + 12$

$\rightarrow 116$

	200
10	0
20	1
30	2
40	3
50	4

arr + index * datatype(size)

$$\text{arr}[0] \rightarrow \text{value}(\text{arr} + 0 \times 4) \\ \Rightarrow \text{value}(200)$$

garbage value adjacent OR error

$$\left\{ \begin{array}{l} \text{arr}[10] \rightarrow \text{value at } (\text{arr} + 10 \times 4) \\ \rightarrow \text{value at } (200 + 40) \\ \rightarrow \text{value at } 240 \end{array} \right.$$

Value at $(\text{base address} + \frac{\text{index} * \text{datatype}}{\text{size}})$

Code:

```
1) int arr[] = {1, 3, 5, 7, 9};  
cout << arr[4] << endl;  
return 0;  
}
```

O/P

9

2) // printing all values

```
int arr[] = {1, 3, 5, 7, 9};  
for (int i=0; i<5; i++) {  
    cout << arr[i] << " ";  
}  
return 0;
```

O/P

1 3 5 7 9

i=4
i<5
print arr[4]

i=5
b < 5 false

3) int arr[10];
 cout << "Enter values for array: " << endl;
 for (int i=0; i<10; i++){ // taking input
 // int n;
 // cin >> n;
 // arr[i] = n;
 // cout >> arr[i];
 // printing
 cout << "printing values in array" << endl;
 for (int i=0; i<10; i++) {
 cout << arr[i] << " ";
 }
 return 0;
 }

```
int arr[10];  

int arr[] = {1, 2};  

for ( )  

{  

    cin >> arr[i];  

}  

for ( )  

{  

    cout << arr[i];  

}
```

arr[0] → value at
 (base + index * data size)
 address

Q. Take 5 element i/p in array & print doubles.

```
int arr[5];
for(int i=0; i<5; i++){
    cin>>arr[i];
}
for(int i=0; i<5; i++){
    cout<<arr[i]*2<< " ";
}
return 0;
}
```

O/P

1
3
5
7
9

~~10 12 14 16~~
1 6 10 14 18

Q. Make $\text{arr}[] = \{1, 3, 5, 7, 9\}$.

\downarrow
 $\text{arr}[0] \text{me} = [0] \text{me}$ $\text{arr}[] \rightarrow \{1, 1, 1, 1, 1\}$
(where taking)

```
int arr[5] = {1, 3, 5, 7, 9};
for(int i=0; i<5; i++){
    arr[i] = i;
}
for(int i=0; i<5; i++){
    cout << arr[i] << " ";
```

H.W

C++ mei main se kya pata hot?

```
int arr[10];
for (int i=0; i<10; i++){
    cout << arr[i] << " ";
}
```

O/P

Garbage value

```
int arr[10] = {1};
for (int i=0; i<10; i++){
    cout << arr[i] << " ";
}
```

O/P

1 0 0 0 0 0 0 0 0 0

Array and functions:-

```
main()
{
    int arr[] = {5, 6};
    inc(arr)
    print array()
}
```

inc (int arr[])

arr[0] = arr[0] + 10

print array()

```

void printArray(int arr[], int size){
    for (int i=0; i<size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

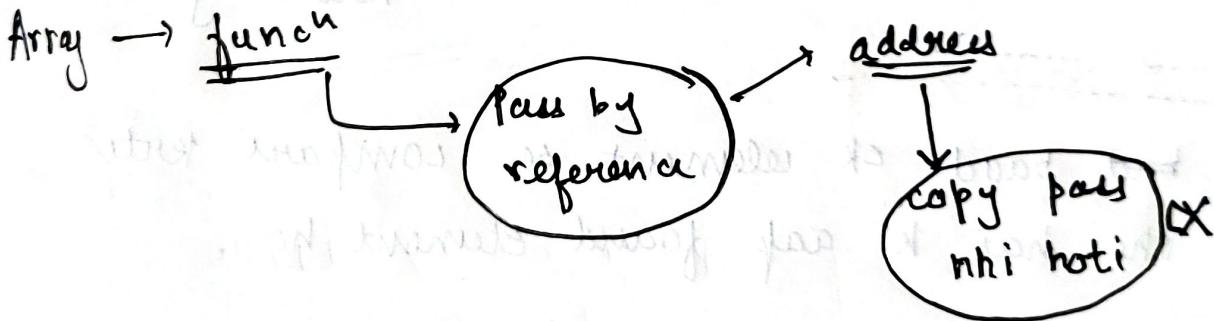
```

```

int main(){
    int arr[] = {5, 6};
    int size = 2;      5 6
    inc(arr, size);
    printArray(arr, size);
    return 0;
}

```

IS 6



→ actual data

be hi open ho
horhe note hain

```

main()
{
    int a[] = {1, 2, 3},
        func(a, size);
    print(a)
}

```

```

func(a[], size)
{
    a[0] = 20;
    a[1] = 30;
    a[2] = 40
    printArray();
}

```

O/P
20 30 40

Linear Search :-

2	1	9	6	7	4	12	15
0	1	2	3	4	5	6	

loop

→ 6 is present or absent in Array → Present ✓

↓
How do you know

ek ~~key~~ baad ek element se compare kerte
ja rhe hai ki aap found element ho .

```

int arr[5] = {1, 3, 5, 7, 8};
int size = 5;
if find(arr, size,
cout << "Enter key to find" << endl;
int key;
cin >> key;
if (find(arr, size, key)) {
    cout << "Found" << endl;
} else {
    cout << "Not Found" << endl;
}

```

```
bool find(int arr[], int size, int key) {
    // linear search
    for (int i=0; i<size; i++) {
        if (arr[i] == key)
            return true;
    }
    // not present
    return false;
}
```

D/P
Enter key to find

5
Found.

2)

```
int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};
int size = 8;
int key = 9;
bool flag = 0;
// linear search

for (int i=0; i<size; i=i+1) {
    if (arr[i] == key) {
        cout << "found" << endl;
        flag = 1;
        break;
    }
}
if (flag)
    cout << "Present" << endl;
else
    cout << "Absent";
}

return 0;
```

Count 0's and 1's in Array

0	0	1	1	0
---	---	---	---	---

3 zero

2 one

```
int arr[] = {0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1};
```

```
int size = 15;
```

```
int numZero = 0;
```

```
int numOne = 0;
```

```
for (int i=0; i<size; i++) {
```

```
    if (arr[i] == 0) {
```

```
        numZero++;
```

```
    } if (arr[i] == 1) {
```

```
        numOne++;
```

```
}
```

```
cout << "numZero" << endl;
```

```
cout << "numOne" << endl;
```

O/P

8

7

for finding no. of 3

→ int count = 0

```
for (int i=0; i<size; i++)
```

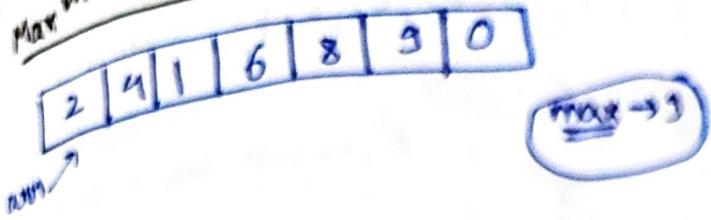
```
{ if (arr[i] == 3)
```

```
    count++;
```

```
}
```

```
cout << count;
```

Max no. in array :-



Best practice

Max no.
→ initialise
INT_MIN

Min no.
→ initialise
INT_MAX

int maxNum = INT_MIN;

```
for(i=0; i<size; i++) {  
    if (arr[i] > maxNum)  
        MAX_NUM = arr[i];  
}
```

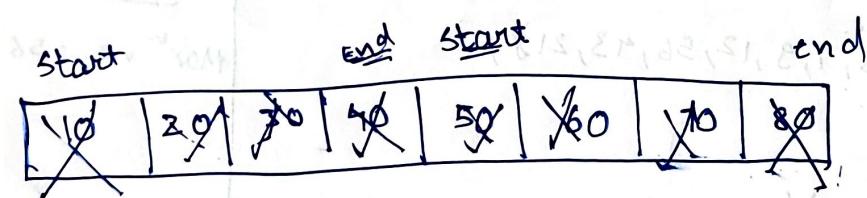
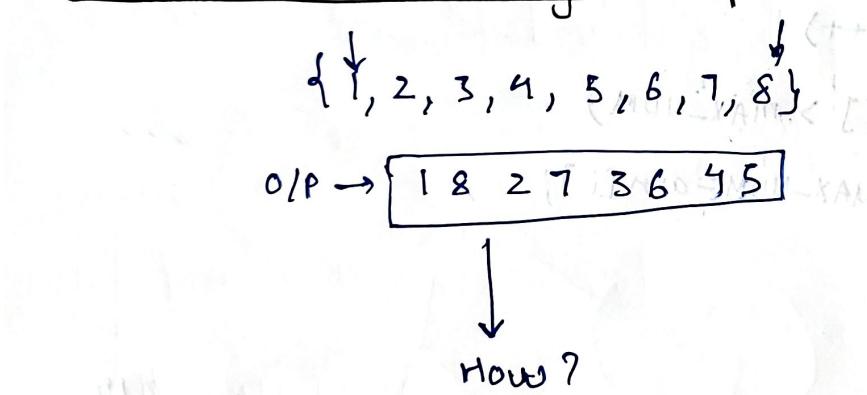
```
#include <iomanip.h>  
int arr[] = {2, 4, 6, 1, 3, 7, 9, 12, 56, 43, 21};  
int size = 11;  
int maxi = INT_MIN;  
for(int i=0; i<size; i++){  
    if (arr[i] > maxi){  
        // found no. greater, update maxi  
        maxi = arr[i];  
    }  
}  
cout << "max no. : " << maxi << endl;  
return 0;
```

O/P
Max no. : 56

Min no. in array :-

```
int arr[] = {2, 9, 6, 1, 3, 7, 9, 12, 56, 43, 21};  
int size = 11;  
int maxi = INT_MIN;  
int
```

Extreme Print in Array :- (2 pointers approach)



↳ print arr[start]

↳ print arr[end]

↳ start++;

↳ end--;

Rukna Rukna

start > end

Rukna Rukna hai

```

int arr[8] = {10, 20, 30, 40, 50, 60, 70, 80};
int size = 8;
int start = 0;
int end = size - 1;

while (true) {
    if (start > end)
        break;

    cout << arr[start] << " ";
    cout << arr[end] << " ";
    start++;
    end--;
}

return 0;
}

```

O/P 10 80 20 60 30 50 40

// given 90 two times for size = 7

```

int size = 7;
int start = 0;

while (true) {
    if (start > end)
        break;

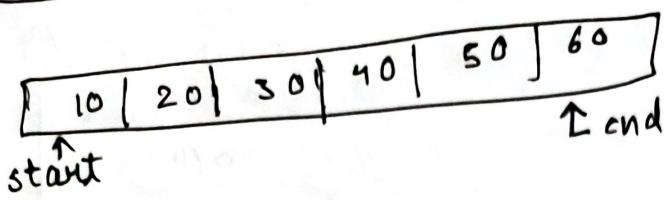
    if (start == end) {
        cout << arr[start] << " ";
    } else {
        cout << arr[start] << " ";
        cout << arr[end] << " ";
    }

    start++;
    end--;
}

```



Reverse an Array:-



60 20 30 40 50 10

60 50 30 40 20 10

60 50 40 30 20 10
end

2 element swap

↳ pre-defined funcⁿ → swap(a, b).

swap(arr[start],
arr[end])

Code:

```
int arr[8] = {10, 20, 30, 40, 50, 60, 70, 80};  
int size = 8;  
  
int start = 0;  
int end = size - 1;  
  
while (start <= end) {  
    // step 1  
    swap(arr[start], arr[end]);  
    // step 2  
    start++;  
    // step 3.  
    end--;
```

↓

// printing

```
for (int i=0; i<size; i++) {  
    cout << arr[i] << " ",  
}
```

O/P

80 70 60 50 40 30 20 10

H.W:

1) Implement swap funcn

↑
swaps

+ -
xor
temp variable

2)

5 Ques kya ein hain, usko code kro

arr 3 baar day run kro.