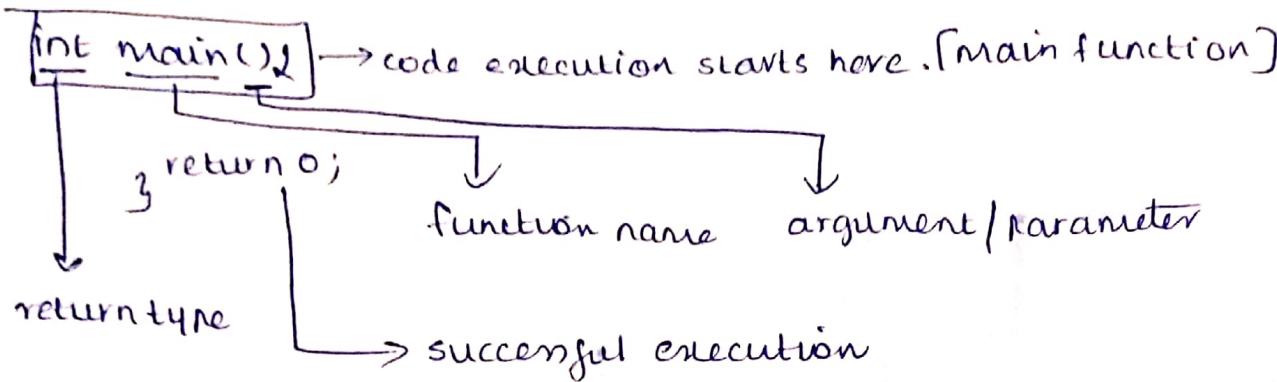


Day 7: Functions & Arrays

23/4/2022

Functions

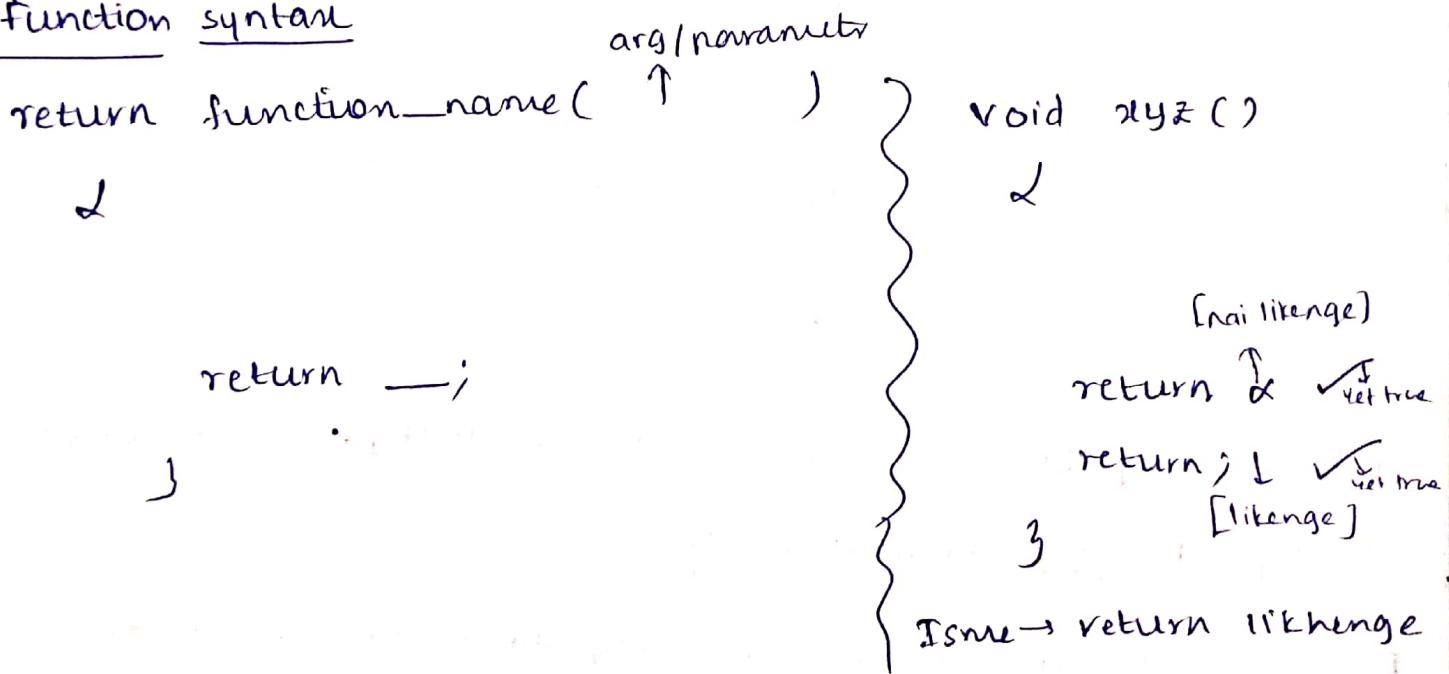
→ Block of code → that performs some operation



NOTE:- I don't want to return anything then → void xyz()

}

Function syntax

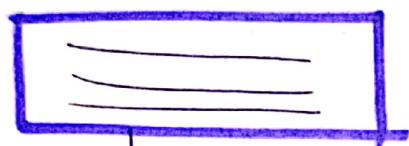


why Function

- ① to avoid repeating of code
- ② readability ↑
- ③ modular approach

- ④ saves time
- ⑤ code clean
- ⑥ maintainability
- ⑦ redundancy
- ⑧ testable
- ⑨ easy to understand

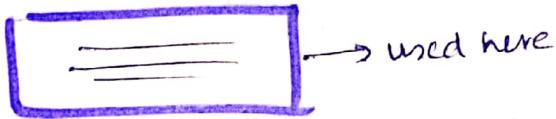
Example:



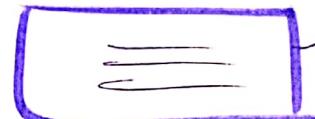
↓
code which ~~is written as~~ is
function repeated in many
times

int main()

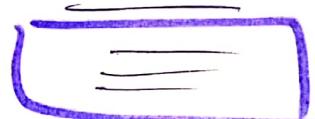
{



→ used here



→ used here



→ used here

}

⇒ Because of same code, used repeatedly, code becomes Bulky, Buggy, Readability]

Example: — ka Har kaam ko, → alog alog module me divide karna and usko → main fn me use karna → [function]

Example:- Function use karne se → code becomes more
Readable.

input()

└
 └
 └

}

Print()

└
 └

Int main()

└ Input()

 Print()

}

// Code 2:-

→ doesn't return anything

void printNumber() {

cout << "Printing number" << endl;

}

int main() {

 int num = 5;

 // function to call:

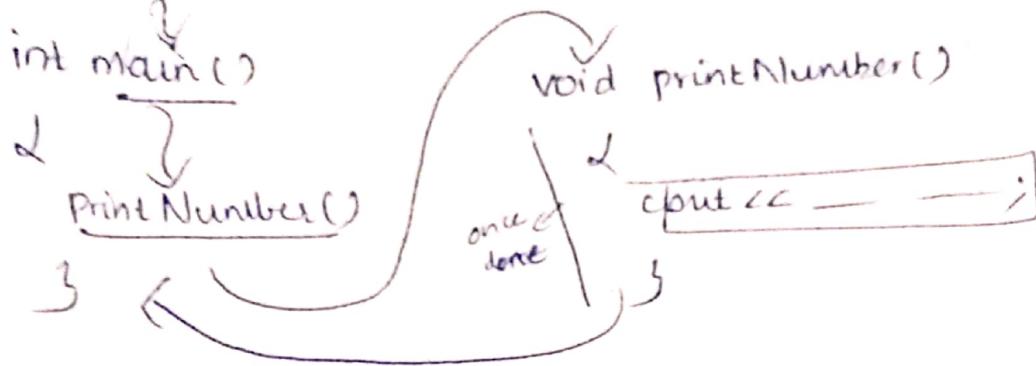
 printNumber();

 return 0;

}

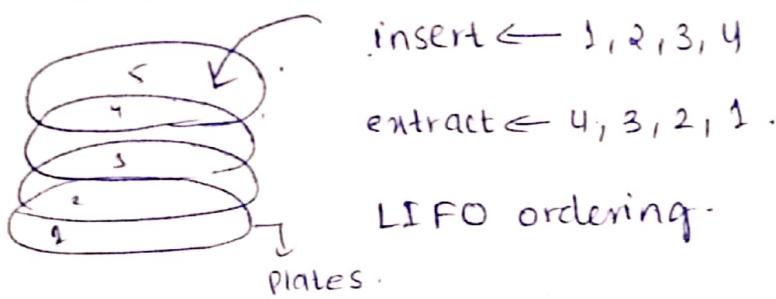
Output: Printing number 0

Dry run:-



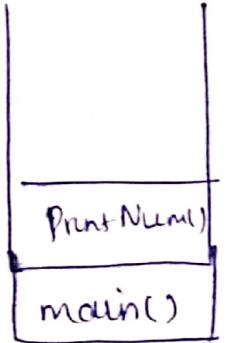
// Function call stack

what is stack?

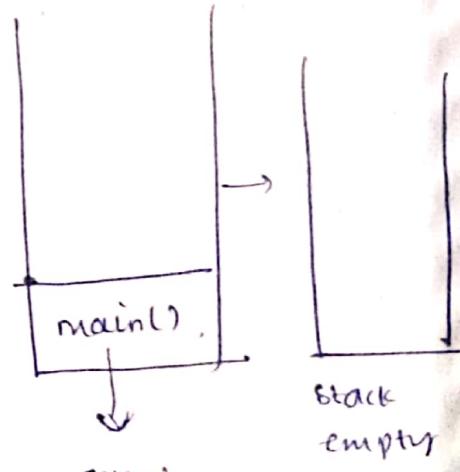


Function call stack →

[Inside memory]



when
it's
done with
execution.
of printNumber()
function.
it comes out



Example:-

main()

arg / fp parameter

1 printNumber(num)

use function no

void printNum(int n)

2 cout << n;

3

NOTE:-

- If parameters → main function me define par [num]
- [PrintNumber / sum printsum] functions will do sum or print number and return it to main().
- num → kousa datatype hai → we have to give it in the function call.

Example:-

printNumber (num), → void printNum(int ⁿ) ;

Code :-

void printNumber (int ⁿ) {

cout << "Printing number" << n << endl;

}

int main () {

int num = 5;

// function to call

Print Number(num);

return 0;

→ ko batana padega,

uske baad integer

aayega so,

}

Pass by value

```

main() {
    int num=5;
    printNum();
    return 0;
}

```

in memory

5

num ↓

block 1

belongs to

```

print(int n) {
    cout << n;
    num++;
}

```

5

num ↓

block 2

3

→ Block 1 & Block 2 are different.

→ Block 2 is copy.

• Block 2 me → If I gonna make any changes.

Ex:- num++ → n=5; ho jayega → n=6.

→ But → Block 1 → value → still remains n=5

→ Block 1 & Block 2 → memory are different.

//code 3:- Pass by value:-

```

void printNum(int num) {
    num=10;
    cout << "Printing number" << num << endl;
}

```

3

int main()

int num=5;

// function to call

printNum(num);

cout << "in main function" << num << endl;

return 0;

↓

O/p:- Printing number 10

in main function 5

B) Approach for and inside memory ka scene → Pass by value

main()

↓

int num=5; block 1
 |
 num

Print(num)
//function call
cout << num
 |
 5
Print ho jaya

Print(int num)

↓

num=10;

cout << num;

↓

block 2
10
num

once the function is completed

block 2 → dies → [coz it was
local variable]

→ then goes back to main fn.

→ MCQs → Pass by value [copy banti hai]

→ Block 1 & Block 2 → memory block → coz int so → [32 bit]
4 byte.

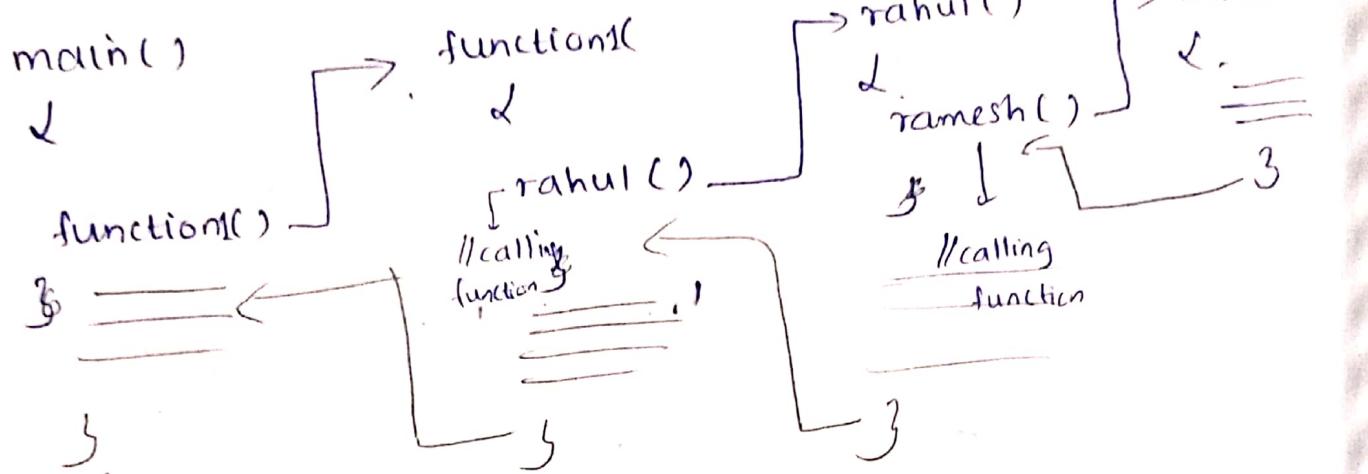
Interviewer → rarely

Online assessment → MCQs → write the O/p

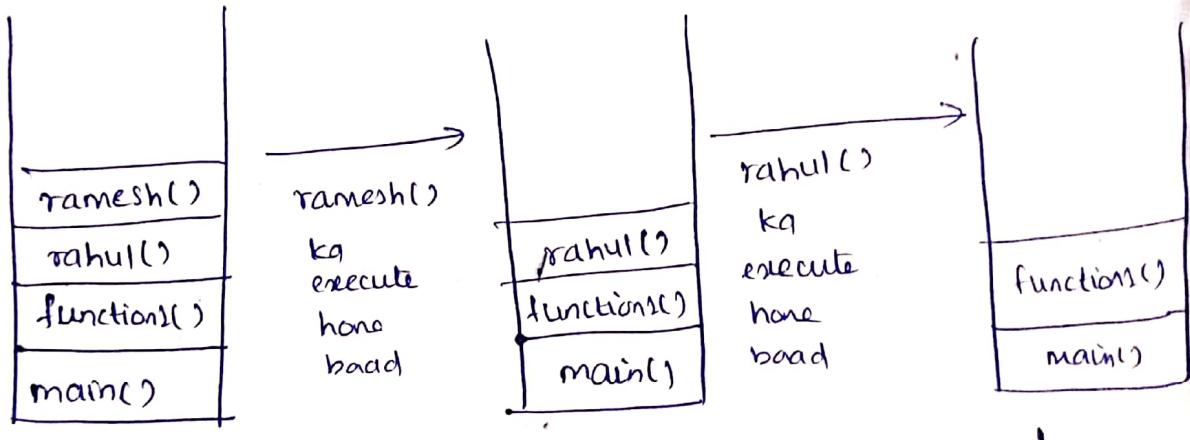
HOMEWORK 12

20+ MCQ's on Pass by value

Example:-



call stack:-



insert → push

extract/remove → pop

execution done

← Stack empty

// code4:- Pass by Value

```
int main()
void ramesh(int n) {
    cout << "ramesh integer" << n << endl;
    n++;
    rahul(n);
}
```

```
void suresh(int k) {
    cout << "suresh integer" << k << endl;
}
```

```
void rahul(int m) {
    cout << "rahul integer" << m << endl;
    m++;
    ramesh(m);
}
```

```
void ramesh(int n) {
    cout << "ramesh integer" << n << endl;
    n++;
    rahul(n);
}
```

```
int main() {
    int n = 21;
```

O(p)

21

22

23

24

Ramesh(n);

cout << "wapas in main function" << endl;

return 0;

}.

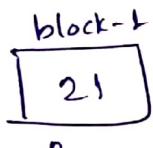
Exploration → int /void main() {
arg → 4 and 4 not

Stack overflow.

Dry run:-

main()

2
 $n = 21$



ramesh(n)

ramesh(n)

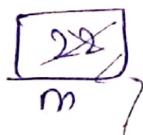
2
cout
 $n++$

rahul(n)

$22 = n$

rahul(int m)

2
cout <<



23

$m++$

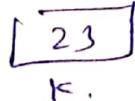
suresh(m)

3

suresh(int k)

2

cout



3.

not printing this

~~21, 22, 23, 24~~, [m=22], [n=22], [rauh ke f' me], [suresh ke f' me], [suresh]

O/P → 21, 22, 23, 24

→ suresh se return hokar, main me jaane par

Question

I/p → given $n = 15$

Print all even no's till n using functions

Code 5

Void printEven(int n) { → approach-1-

 for (int i=2; i<=n; i=i+2) {

 cout << i << " ";

}

}

int main() {

 int n=22;

 printEven(n);

 return 0;

}

O/p:- 2 4 6 10 12 14 16 18 20

approach-2

For (int int i=1; i<=n; i++)

 if (i%2==0)

 cout << i

} or

~~if (i&1) is (!(i&1))~~ → odd

cout << i → even

}

checking if even

or not

Qn) Given $n=6$ $\rightarrow 1, 4, 9, 16, 25, 36$
Print all squares till n.

//code 6:- Printing squares till n:-

void printSquares(int n)

2.

```
for (int i=1; i<=n; i++) {  
    cout << i*i << ",";
```

}

}

int main() {

int n=6;

printSquares(n);

return 0;

}

}

②

①

NOTE:-

if we write code block ② after code block ① we get.

Error: use of undeclared Identifier]

so, function call, humesha, call karne se pehle hona chahiye

Q) IIP → Given $n=5$

Factorial of a number provided → IIP → $5! = 120$.

//code

$$5! = 5 \times 4 \times 3 \times 2 \times 1.$$

→ coz, we are returning number (i.e., factorial of a num)

```
int getFactorial(int n) {
```

```
    int ans = 1;           → //not 0, coz, multiply kar-hai  
    for (int i=n; i>=1; i--) {
```

$$ans = ans * i;$$

}

```
    return ans;
```

}

```
int main() {
```

```
    int fact = getFactorial(5);
```

inplace of 0, 1, → bhi run hata hai

```
    cout << fact << endl;
```

inplace of 5 → 12. ka factorial.

```
    return 0;
```

correct

}

NOTE: this code is running for 1 - 12 factorial only

13, 14, 15 → ✗ → wrong ans

coz, int ka value overflow hogा.

[To overcome → we use arrays]

Q) If $n=10$ given

Create an array to print Name.

Arrays → what?

→ It's a ds → ~~any~~ kind of data.
same

[It stores in continuous
memory location]

may it be int/char/string/customer object

Example:-

$n=3$

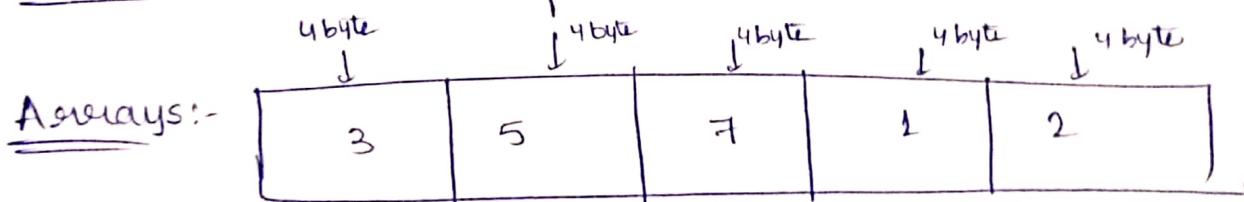
A B C

find the maximum number.

→ approach: - what we studied
① ↳ If-cond^n.

If $n=30 \rightarrow$ app ① not possible.

NOTE:- when number ↑ approach fails.



int a=5

5
a

Continuous memory location.

int arr[5];

int type ka data.

size

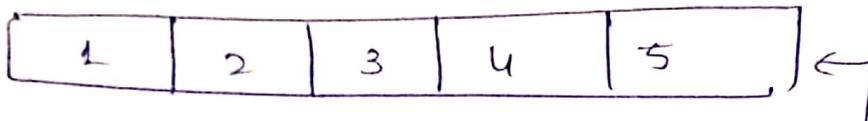
total size = 20
byte.

arr [100]
[Address of
first block]

base address

Qn:- I want to create an array and store 1, 2, 3, 4, 5

int arr[5] = {1, 2, 3, 4, 5} initialize -
↓ [sc statement se ye bana] ①



int arr[] = {1, 2, 3, 4, 5}; ②

① & ② are same, they are array creation.

int arr[5]; → we are not providing values;
so, pehle ke garbage values rahenge

Code 8:-

array creation

```
int main() {  
    // creation  
    int arr[10];  
    return 0;  
}
```

Code 9:-

int main() {

```
    int arr[5] = {1, 2, 3, 4, 5};  
    return 0;  
}
```

Code 10:-

```
int main()  
{  
    int arr[] = {1, 2, 3, 4, 5};  
    return 0;  
}
```

Ques 11:-

```
int main() {
```

```
    int arr[5] = {1, 2};
```

```
    return 0;
```

```
}
```

5 ≠ no of
elements.

This also gets executed.

Ans:-

```
int arr[5] = {1, 2}.
```



garbage values or 0

→ dependent on compiler.

→ I want to initialize entire array with single value.

Poor arrays me 0 zero daane

```
int arr[5] = {0}; → 0 se initialize ho Jayega.
```

→ can I initialize entire array with 1.

```
int arr[30] = {1}
```

Homework:-

// accessing the array values
arr

100	104	108	112	116.
3	4	9	7	2.
0	1	2	3	4

→ through index → we can access the elements of array.

$$\text{1}^{\text{st}} \text{ element} \rightarrow \text{arr}[0] = 3 \quad \text{arr}[1] = 4 \\ \text{arr}[3] = 7 \quad \text{arr}[4] = 2$$

approach:- $\text{arr}[0]$
(Behind the
scene)

$$100 + 0 \times 4 = 100 \rightarrow \text{pc 3 hai element}$$

$$\text{arr}[3] = 100 + 3 \times 4 = 112 \rightarrow \text{pc 7 hai element}$$

$$\boxed{\text{arr}[i] = \text{base address} + i \times \text{size of data}} \quad \text{where } i \text{ is index}$$

Qn:- create create fn, and which prints elements of an array

// code 12:- [accessing the elements]

int main() {

int arr[5] = {3, 6, 9, 2, 18}

with or without size .i.e., arr[]

void printArray (int arr[5], int size) {

for (int i=0; i < size; i++) {
// accessing array's elements
cout << arr[i] << " ";

must in arrays

}

}

```
int main() {
```

```
    int arr[5] = {3, 6, 9, 2, 18};
```

```
    PrintArray(arr, 5);
```

```
    return 0;
```

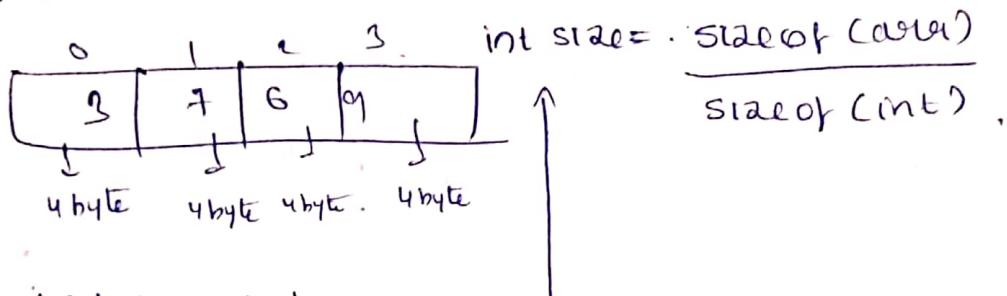
[size of array (imphai, varna
error ayega)]

3.

NOTE:-

sizeof → operator

find the size of the arr[5] = {1, 2, 3, 4, 5} using above operator.



16 byte total
poora kitni memory le raha hai
2 box ki memory.

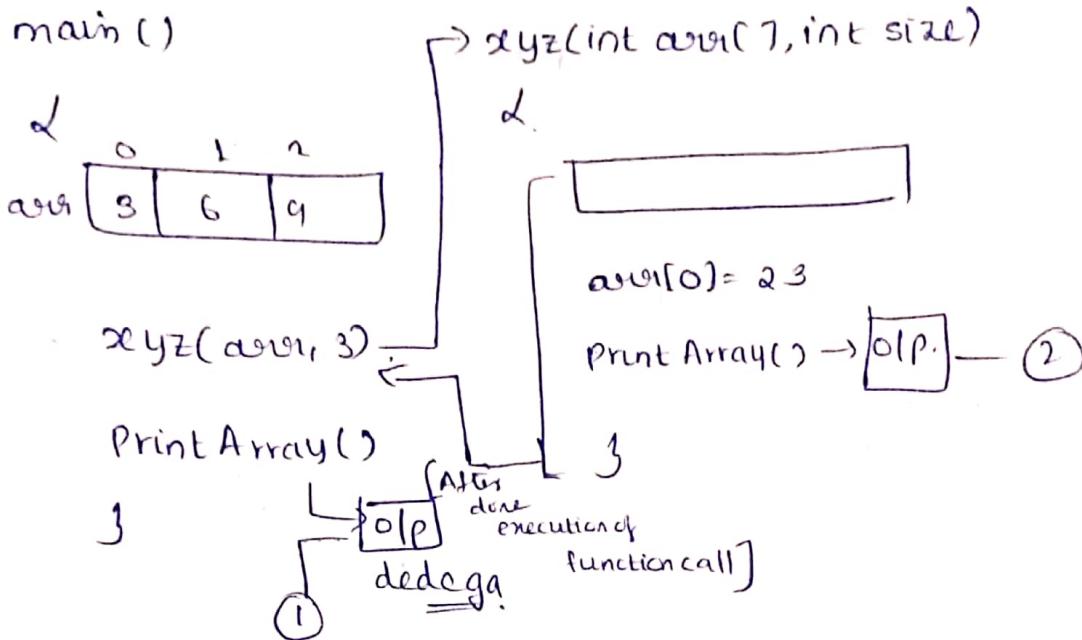
But it's not for all cases

Example:- int arr[30] = {1, 2, 3}.

$$\text{int size} = \frac{\text{sizeof}(\text{arr})}{\text{sizeof}(\text{int})} \quad \text{ans} = 30 \times$$

③ → elements hi hai, 30 not needed.

Using array with a function



//code :-

```
void util(int arr[], int size) {  
    arr[0]=23;  
    cout<<"printing in UTIL function"<<endl;  
    printArray(arr, 3);  
}  
  
int main() {  
    int arr[] = { 3, 6, 9 };  
    util(arr, 3);  
    cout<<"printing in main function"<<endl;  
    printArray(arr, 3);  
    return 0;  
}
```

OLP 23 6 9 } → OLP same ① & ② are same

23 6 9 }

→ ~~It's~~ diff in [pass by value]
olp is.

→ In using array with function → olp is same

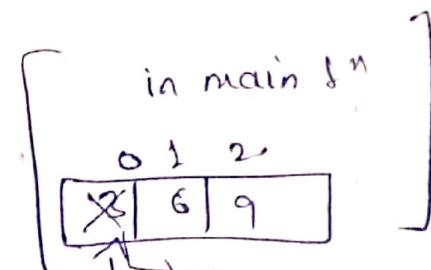
↳ When we are passing array in function and enter
in the function [say util]

[array is not copied]
[util fn]

→ only $\text{arr}[0] = 23$

$$100 + 0 \times 4 = 100 \rightarrow \text{uspaat jadak}$$

23 daaldo.



main()

$\text{arr}[] = \{1, 2, 3, 4\};$

$\text{xyz}();$

return arr;

and comeback

to main fn
after updation.

f^{n:-}
 $\text{xyz}();$

L.

arr value
when we update
here

3

main fn will sustain the change, no anywhere

Q1:- linear search:-

arr	3	6	7	12	2	4	6
	0	1	2	3	4	5	6

check $\boxed{\text{target} = 4}$

approach:- using loop, \rightarrow when reached \rightarrow return +/F

//code 34:- linear search

bool search(int arr[], int size, int target).

int main() {

// [int arr[n];] \rightarrow // Bad practice-

int arr[100];

cout << "Enter the no of elements" << endl;

int n;

cin >> n;

// taking input in array

for (int i=0; i<n; i++) {

cin >> arr[i];

}

cout << "Printing the elements in Array" << endl;

printf("%d", n);

array

return 0;

}

// code approach :-

```
bool search(int arr[], int size, int target) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == target) {
            return true;
        }
    }
    return false;
}

int main() {
    int arr[100];
    cout << "Enter the num of elements" << endl;
    int n;
    cin >> n;
    // taking input in array
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    if (search(arr, 5, 7)) {
        cout << "Element found" << endl;
    } else {
        cout << "Element not found" << endl;
    }
    cout << "Printing elements in array" << endl;
    PrintArray(arr, n);
}
}
```

W/o Prolog