1) Laptop / PC

2) Dedication

C++ Programming in One Shot

Chapter 1 : Basics

1) Basic Printing, \n and endl wali cheez.
2) Variables, printing variables, int, float, and +,-,*,/ of integers.
3) Variable naming rules.
4) Comments
5) Taking Input
6) Modulus Operator
7) Float to int, int to float
8) Hierarchy
9) Char and ASCII

Basic program in C++

```
#include<iostream>
using namespace std;
int main(){
    cout<<“hello world”;
}
```

Output

Hello World

How to move in next line?

Example :

```
cout<<"Hello PW";

cout<<"Hello CW";
```

Output will be :

```
Hello PWHello CW
```

'DRY RUN'

```
1  #include<iostream>
2  using namespace std;
3  int main(){
4      cout<<"Hello PW";
5      cout<<endl;
6      cout<<endl;
7      cout<<"Hello CW";
8  }
```

Output

- Hello PW
-
- Hello CW

Use of escape sequence endl and '\n'

Example :

```
cout<<"Hello PW";
cout<<endl;
cout<<"Hello CW";
```

Output will be :

```
Hello PW
Hello CW
```

* A good thing :

1) cout << 4;

2) cout << "4";

3) cout << 4+3;

4) cout << "4+3";

Variables and their Declaration

Boxes, Dabbe

Let us focus on int data type as of now.

1) Variables as containers :

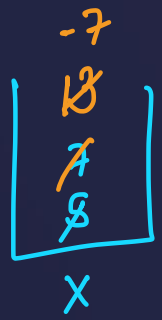
```
#include<iostream>
using namespace std;
int main(){
    int x;
    x = 5;
    cout<<x;
}
```

Output

5

Printing Variables in C & Updation of Variables

```
int x = 5;
cout<<x<<endl;
x = 7;
cout<<x<<endl;
x = x + 6;
cout<<x<<endl;
x = x - 20;
cout<<x<<endl;
```



$x = x - 20$

$x = 13 - 20$

$= -7$

Output

- 5
- 7
- 13
- -7

```
int x = 5;
cout<<x<<endl;
x = 8;
cout<<x<<endl;
```

Updation

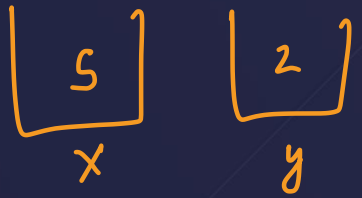
Output

- 5
- 8

Arithmetic operations on int data type

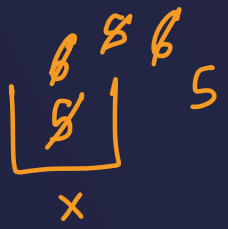
```
int x = 5;
int y = 2;
cout<<x+y<<endl;
cout<<x-y<<endl;
cout<<x*y<<endl;
cout<<x/y<<endl; // issue
```

int int



Increment - Decrement operators

```
int x = 5;
x++;
cout<<x<<endl;
x--;
cout<<x<<endl;
++x;
cout<<x<<endl;
--x;
cout<<x<<endl;
```



Thodi der me

Output

6
5
6
5

Post Increment

```
int x = 4;
cout<<x<<endl;
cout<<x++<<endl;
cout<<x<<endl;
```

Output

- 4
- 4
- 5

Pre increment

```
int x = 4;
cout<<x<<endl;
cout<<++x<<endl;
cout<<x<<endl;
```

Output

- 4
- 5
- 5

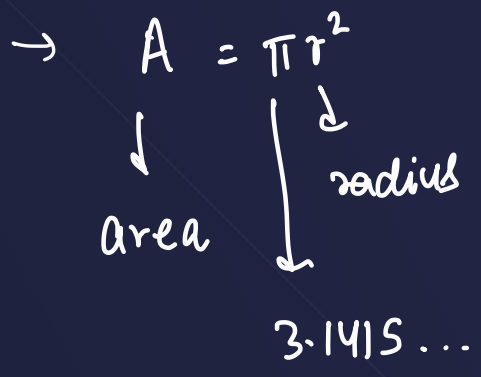
float data type → Storing Real Numbers

```
float x = 3.1;
```

Arithmetic operations on float data type

```
float x = 5;

float y = 2;

cout<<x+y<<endl;

cout<<x-y<<endl;

cout<<x*y<<endl;

cout<<x/y<<endl;
```

Example : Calculating Area of a Circle



Example : Calculating Simple Interest

$$SI = \frac{PRT}{100}$$

```
float p;
float r;
float t;
float si;
```

p = 540

r = 32

t = 3

$$\frac{540 \times 32 \times 3}{100} = \frac{5184}{10}$$

$$= 518.4$$

SI = 518.4

Homework : Calculate Volume of a Sphere

Variable Naming rules

1) Variables can start from an alphabet or underscore _ or $.
2) Special characters except _ and $ are not allowed.
3) Some particular keywords are not allowed.
4) Commas or blanks are not allowed.

auto	double	int	break	extern	enum	unsigned	while
case	sizeof	for	const	static	long	continue	float
else	signed	do	short	switch	char	volatile	default
goto	struct	if	union	return	void	register	typedef

Variable Naming rules - Examples

Q. Which of the following are invalid variable names and why?

BASICSALARY	_basic	basic-hra
#MEAN	group.	422
population in 2006	over time	mindovermatter
FLOAT	hELLO	queue.
team'svictory	Plot#3	2015_DDay

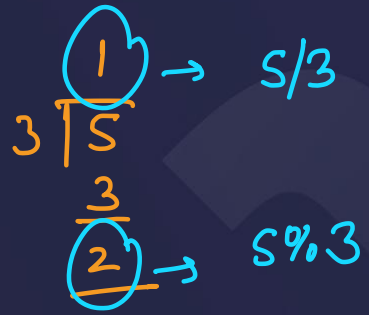
Taking input // Square of a Number

Take 2 numbers input from user and print their Sum

Modulus Operator

a%b is remainder when a is divided by b

+, -, *, /, %



→ x = 5

→ y = 3

x+y = 8

x-y = 2

x*y = 15

x/y = 1

x%y = 2

Modulus Operator → importance is divisibility

2 % 8 = 2

Some important points :

1) $a \% b = a$ [if $a < b$]

2) $a \% a = 0$

3) $a \% (-b) = a \% b$

4) $(-a) \% b = -[a \% b]$

Typecasting → float → int, int → float

Ques : Take integer ‘x’ as input and print half of the number.

```
int x;
cin >> x;
```

Hierarchy of operators → very important

(BODMAS)

B{ O{ D,M | A,S

Left to Right

```
int i = 2 * 3 / 4 ;
cout<<i;
```

Maths

$i = 2 * 3/4$

$= 2 * 0.75$

$= 1.5$

$6/4$

$= 1.5$

C++ (int)

$i = 2 * 3 / 4$

$= 6/4$

$= 1$ ✓

$i = 2 * 3/4$

$= 2 * 0$

$= 0$ ✗

char data type

```
char ch = 'a';
```

↳ a, b, c ... z

A, B, C ... Z

!, @, #, $, %, ^, &, *, (,), {, }, ', ", -, _, +, =, [,], |, /, \

., , , ?, /, >, :, <, ~, `

0, 1, 2, 3 ... 9

ASCII values → very important [Typecasting]

```
char ch = 'a';
```

'a' → 97
b → 98
c → 99
d → 100
⋮
z → 122

A → 65
B → 66
C → 67
⋮
Z → 90

'0' → 48
'1' → 49
⋮
'9' → 57

MCQ Time !

Homework

In b = 6.6 / a + 2 * n ; which operation will be performed first?

(1) 6.6 / a

(2) a + 2

(3) 2 * n

(4) Depends upon compiler

MCQ

Which of the following statements is false

(1) Each new C++ instruction has to be written on a separate line FALSE

(2) Usually all C++ statements are entered in small case letters TRUE

(3) Blank spaces may be inserted between two words in a C++ statement TRUE

(4) Blank spaces cannot be inserted within a variable name TRUE

C. W.

~~Homework~~

float

The expression, a = 7 / 22 * (3.14 + 2) * 3 / 5 ; evaluates to

(1) 8.28

(2) 6.28

(3) 3.14

(4) 0 ✓

→ a = 7/22 * 5.14 * 3 / 5

⇒ a = 0 * 5.14 * 3 / 5

⇒ a = 0 * 3 / 5

⇒ a = 0 / 5

⇒ a = 0

V. Important:

```
float x = 5/2;
```

int (5/2)

int (float x)

real no.s ko accept

5/2 = 2

5.0/2 = 2.5

5/2.0 = 2.5

5.0/2.0 = 2.5

MCQ

/, *, % > +, -

The expression x = 4 + 2 % - 8 evaluates to

⇒ x = 4 + 2 = 6

(1) -6

(2) 6

(3) 4

(4) None of the above

2 % - 8 = 2 % 8 = 2

a % (-b) = a % b

a % b = a [a < b]

**MCQ → Homework

What will be the value of d if d is a float after the operation d = 2 / 7?

(1) 0

(2) 0.2857

(3) Cannot be determined

(4) None of the above

Chapter 2 : Conditionals

1) if, if - else
2) nested
3) Else if ladder
4) Ternary
5) switch

ans/output

situation dependent

IF – ELSE

अगर मगर वाली बातें

Ques : Take positive integer input and tell if it is even or odd

```
int n;
cin>>n;
```

1) cout << "Even";
2) cout << "Odd";

even

$\frac{n}{2}$ → rem = 0

if (n % 2 = 0) even

n % 2 ≠ 0 odd

→ a > b

a == b , a != b

```
int n;
cout<<"enter n : ";
cin>>n;
if(n%2==0) cout<<"even number";
if(n%2!=0) cout<<"odd number";
```

Output

Enter n : 3

Odd number

HW : Take positive integer input and tell if it is divisible by 5 or not.

Ques : Take integer input and print the absolute value of that integer

+, -, *, /, % → Arithmetic op

```
int n;
cin>>n;
```

7 → 7

10 → 10

-21 → 21

==, !=, >, <, >=, <= → relational op

```
if (n>=0) → cout<<n
else → cout<<-n;
```

*Ques : If cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss or no profit no loss. Also determine how much profit he made or loss he incurred.

```
int cp;
cin >> cp;

int sp;
cin >> sp;
```

if (sp > cp) profit → sp - cp

if (cp < sp) loss → cp - sp

if (cp == sp) no profit no loss

HW : Given the length and breadth of a rectangle, write a program to find whether the area of the rectangle is greater than its perimeter.

l, b

↓

A = l * b;

P = 2 * (l + b);

Multiple Conditions Using && and ||

and

or

Ques : Take positive integer input and tell if it is a three digit number or not.

if (n > 99 and n < 1000) cout << —

n > 99 → true

n < 1000 → true

Ques : Take positive integer input and tell if it is divisible by 5 and 3.

```
if ( n % 5 == 0   &&   n % 3 == 0)
```

Ques : Take positive integer input and tell if it is divisible by 5 or 3.

→ 3, 6, 9, 12, 15, 18, 21, 24, 27, 30 . . .

→ 5, 10, 15, 20, 25, 30, 35. . .

↳ 3, 5, 6, 9, 10, 12, 15, 18, 20, 21, 24, 25, 27, 30,

if (n%5==0 || n%3==0) —

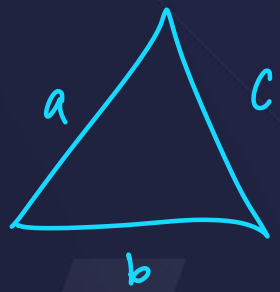
n = 82

Ques : Take 3 numbers input and tell if they can be the sides of a triangle.

a, b, c

&&, and

if ((a+b) > c and (c+a) > b and (c+a) > b)



a + b > c

b + c > a

c + a > b

Ques : Take 3 positive integers input and print the greatest of them.

a, b, c

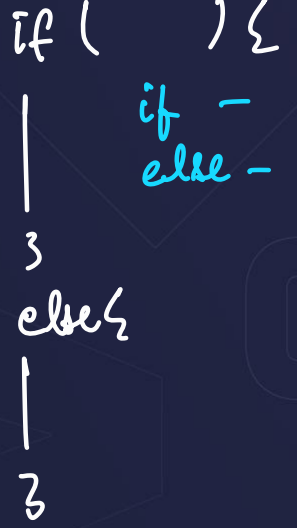
```
if ( a > b  and  a > c )   cout << a ..
else if ( b > a  &&  b > c )  cout << b
else  cout << c
```

Homework : Take 3 positive integers input and print the least of them.

HW : Take positive integer input and tell if it is divisible by 5 or 3 but not divisible by 15.

```
if ((n%5==0 or n%3==0) and (n%15!=0))
```

Nested If – Else



Ques : Take 3 positive integers input and print the greatest of them. without using multiple conditions

HW : If the ages of Ram, Shyam and Ajay are input through the keyboard, write a program to determine the youngest of the three.

Else If

Ques : Take input percentage of a student and print the Grade according to marks:

1) 81-100 Very Good
2) 61-80 Good
3) 41-60 Average
4) <=40 Fail

HW : Given a point (x, y), write a program to find out if it lies in the 1st Quadrant, 2nd Quadrant, 3rd Quadrant, 4th Quadrant, on the x-axis, y-axis or at the origin, viz. (0, 0).

if-else but cool banne ke liye lines bachana

Ternary Operator

expression 1 ? expression 2 : expression 3 ;

another replacement for if-else

Switch Statement

Useless

Syntax :

character

```
switch ( integer expression ) {
    case constant 1 :
        do this ;
    case constant 2 :
        do this ;
    case constant 3 :
        do this ;
    default :
        do this ;
}
```

int

n1, n2, op

int char

Raghav's Recommendation : Exam se 1 day before

Ques : Write a program to create a calculator that performs basic arithmetic operations (add, subtract, multiply and divide) using switch case ~~and functions~~. The calculator should input two numbers and an operator from user.

```
int a;
int b;
char ch; → +, -, /, *
```

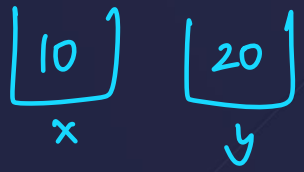
→ a ch b

→ n1 op n2

MCQ Time !

Predict the output

```
main( ) {
    int x = 10, y = 20 ;
    if ( x == y ) ;
            cout<<x<<" "<<y ;
}
```



Output

10 20

HW : Predict the output

```
int main( ) {
    int x = 3, y = 5 ;
    if ( x == 3 )
        [illegible] cout << x <<endl;
    else ;
        [illegible] cout << y << endl;
}
```

*Predict the output

```
main( ) {
    int x = 3, y, z ;
    y = x = 10 ;
    z = x < 10 ;
    cout<<x<<" "<<y<<" "<<z ;
}
```

(Right to Left)

true → 1

Output

10 10 0

Chapter 3 : LOOPS → repetition

1) For Loop
2) Break and Continue
3) While and Do While Loop
4) Questions using Operators
5) Pattern Printing Problems

What and Why?

For Loop

```
for(int i = 1; i<10; i++){

    // code

}
```

condition

increment/ decrement

iterator

```
int x = 3;
int i = 1;
```

declaration + initialization

Ques : Print hello world ‘n’ times. Take ‘n’ as input from user

Done!

How for loop works : the various parameters.

n=3

```
for(int i=1;i<=n;i++){
    cout<<"Hello World"<<endl;
}
```

Output

- Hello World
- Hello World
- Hello World
- .
- .

Initialization

Steps :

1) Check condition
2) Work
3) Increment

Ques : Print numbers from 1 to 100

Using for loop

Ques : Print all the even numbers from 1 to 100

if (i % 2 == 0) → even

Method-2 :

HW : Print all the odd numbers from 1 to 100

Ques : Print the table of 19.

1

19, 38, 57, 76, 95, 114, 133, 152, 171, 190

HW : Print the table of ‘n’. Here ‘n’ is a integer which user will input.

Ques : Display this AP – 1,3,5,7,9.. upto 'n' terms.

1 (+2) 3 (+2) 5 (+2) 7 (+2) 9

Method-1 : n^{th} term formula

$\rightarrow a_n = a + (n-1)d$

$\Rightarrow a_n = 1 + (n-1)2$

$\Rightarrow a_n = 1 + 2n - 2$

$\Rightarrow a_n = 2*n - 1$

4, 7, 10, 13. . . .

Ques : Display this AP – ~~1,3,5,7,9~~.. upto 'n' terms.

rounds

Method-2 : Using extra variable & keeping 'i' only for the iterations

HW : Display this AP - 4,7,10,13,16.. upto 'n' terms.

$a = 4, \quad d = 3$

$a_n = a + (n-1)d$

$a_n = 4 + (n-1) \cdot 3$

$a_n = 4 + 3n - 3$

$a_n = 3n + 1$

$a_n = 3*n + 1$

Ques : Display this GP – 1,2,4,8,16,32,.. upto 'n' terms.

x2 x2 x2 x2

Ex:

→ 5, 15, 45, 135,

```
for(int i=1; i<=n; i++){
    cout << a << " ";
    a = a*2;
}
```

HW : Display this GP - 3,12,48,.. upto ‘n’ terms.

Break;

I

thodi der me

ques ke saath

Ques : WAP to find the highest factor of a number 'n' (other than n itself)

24 → 1, 2, 3, 4, 6, 8, 12, 24

n

1 to n-1

```
int f = 1; //f me factor
for (int i = 1; i < n; i++){
|  if(n%i == 0) f = i;
3
```

Ques : WAP to check if a number is composite or not.

if n has extra factor/factors except 1 & n then it is composite

↓

if (n%i==0) → we found a factor

↓

loop → 2 to n-1 | 2 to n/2

Ques : WAP to check if a number is prime or not.

Done

Continue;

↓

if you want to skip a round

↓

Print no. from 1 to 20 except 3 & 8

H.W.

Ques : WAP to print odd numbers from 1 to 100. using continue

While Loop → An alternate to for loop

Kitni baar chalega, weird scenario

multiple conditions

```
int i = 0;
while(i<10){
    // code
    i++;

}
```

Do- While Loop

→ ek baar kam se kam run kare

```
do {

    //code

} while ( another == 'y' ) ;
```

Predict the output

```
int main( ) {
    int x = 1 ;
    while ( x == 1 )
    x = x - 1 ;
    cout<<x<<endl;
}
```

Output

0

Predict the output

```
main( ) {
    int i ;
    while ( i = 10 ) {
        cout<<i<<endl;
        i = i + 1 ;
    }
}
```

→ Infinite Loop

Predict the output

```
main( ) {
    while ( 'a' < 'b' )
        cout<< “malayalam is a palindrome" <<endl ;
}
```

(97 < 98)

↓

Infinite loop

HW : Predict the output

```
main( ) {
    int i = 10 ;
    while ( i = 20 )
    printf ( "\nA computer buff!" ) ;
}
```

HW : Predict the output

```
main( ) {
        int x = 4, y = 0;
        while ( x >= 0 ) {
                x-- ;
                y++ ;
                if ( x == y )
                        continue ;
                else
                        cout<<x<<" "<<y<<endl;
        }
}
```

Questions using Operators

+ - * / %

Ques : WAP to count digits of a given number.

→ int n;
cin>>n;

#Hint : 1) Integer division
2) Divide n by 10 again & again

n = 5 8 1 2

```
int count = 0;
while (n > 0) {
    n = n/10;
    count++;
}
```

Ques : WAP to count digits of a given number.

n = 1234

n = 123

n = 12

n = 1

n = 0

finish

count = 0

count = 1

count = 2

count = 3

count = 4

ans

Ques : WAP to print sum of digits of a given number.

n = 1234 Sum = 1+2+3+4 = 10 ✓

Hint:

1) Modulus (%) operator
2) n % 10 gives the last digit
3) 1+2+3+4 = 4+3+2+1

n = 51284 3

Ques : WAP to print sum of digits of a given number.

1+2+3+4 = 4+3+2+1

ld = n%10

Algorithm :

n	ld	sum
1234	4	4
123	3	7
12	2	9
1	1	10
0		

loop finished

HW : WAP to print product of digits of a given number. (There won't be a zero in the number)

For Ex: n = 1234

pro = 1 * 2 * 3 * 4 = 24

HW : WAP to print sum of all the even digits of a given number.

$n = 1234$

$\Rightarrow \quad 2 + 4 = 6$

Ques : WAP to print reverse of a given number.

n = 1234 → r = 4321

Hint : 1) Sum of digits

2) 4321 = 0, 0, 4, 40, 43, 430, 432, 4320, 4321

Ques : WAP to print reverse of a given number.

Algorithm :

$r = r * 10$

$r = r + ld$

n = 1234 123 12 1 0

r = 0 0 4 40 43 430 432 4320 4321

ld = 4 3 2 1

1) ld
2) r *= 10
3) r += ld
4) n /= 10

HW : WAP to print the sum of given number and its reverse.

n = 1234 , r = 4321

Ques : Print the factorial of a given number 'n'.

→ Find the sum from 1 to 'n' → $\frac{n*(n+1)}{2}$

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

$8! = 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$

$n! = n \times (n-1) \times (n-2) \cdots 3 \times 2 \times 1$

or

$1 \times 2 \times 3 \times 4 \cdots (n-1) \times n$

HW : Print the factorials of first 'n' numbers

#Hint : n!

n = 5,

1! = 1

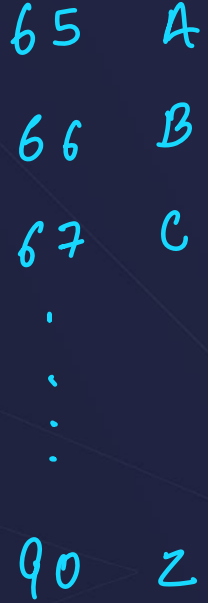
2! = 2x1 = 2

3! = 3x2x1 = 6

4! = 4x3x2x1 = 24

5! = 5x4x3x2x1 = 120

HW : Write a program to print all the ASCII values and their equivalent characters of 26 alphabets using a while loop.



Chapter : 4

Pattern Printing Questions

↓

Nested Loops

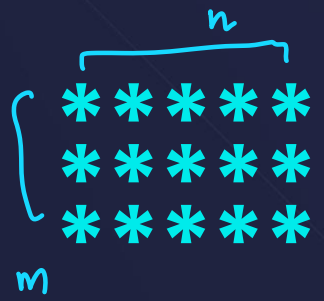
↓

Nested if-else

↓

ek ke andar ek

Ques : Print the given pattern



m x n ka star rectange

m = 2

n = 6

Solid Rectangle

Ques : Print the given pattern

Solid Square

Ques : Print the given pattern

n = 4

```
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
```

n = 3

```
1 2 3
1 2 3
1 2 3
```

Number Square

Homework :

n = 4

1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4

n = 2

1 1
2 2

n = 5

1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5

Ques : Print the given pattern

n=4

A B C D
A B C D
A B C D
A B C D

n= 3

A B C
A B C
A B C

Alphabet Square

HW : 1

A A A A
B B B B
C C C C
D D D D

HW : 2

a b c
a b c
a b c

(n = 3)

Ques : Print the given pattern

```
    1 2 3 4  → j
1   *
2   **
3   ***
4   ****
|
i
```

i = 1, j = 1

i = 2, j = 1, 2

i = 3, j = 1, 2, 3

i = 4, j = 1, 2, 3, 4

j <= i

Star Triangle

Ques : Print the given pattern

H.W.

```
1
1 2
1 2 3
1 2 3 4
```

→ Number Square code → j<=n ↓ j<=i

Number Triangle

HW : Print the given pattern

```
A
A B
A B C
A B C D
```

Alphabet Triangle

HW :

1
2 2
3 3 3
4 4 4 4

i → row no.
j → col no.
n → no. of rows
m = no. of cols

x x x 1 2 3 1 1 1
x x x 1 2 3 2 2 2
x x x 1 2 3 3 3 3

A B C
A B C → j<=i
A B C

*HW : Print the given pattern

n = 5

```
1
A B
1 2 3
A B C D
1 2 3 4 5
```

n = 3

```
1
A B
1 2 3
```

Hint → if else

Alphabet Triangle

Ques : Print the given pattern

```
i  1 2 3 4 → j
1  * * * *
2  * * *
3  * *
4  *
```

$i = 1,\ j = 1, 2, 3, 4$

$i = 2,\ j = 1, 2, 3$

$i = 3 \rightarrow j = 1, 2$

$i = 4, \rightarrow j = 1$

$\rightarrow \quad i + j_{max} = n+1$

$\Rightarrow \quad j_{max} = n+1-i$

$\Rightarrow \quad \boxed{j <= n+1-i}$

```
1  X
2  X X
3  X X X
4  X X X X
```

$\boxed{j <= i}$

Star Triangle Ulta

HW : Print the given pattern

```
1 2 3 4
1 2 3
1 2
1
```

and

H·W·

```
1 1 1 1
2 2 2
3 3
4
```

and

H·W·

```
4
4 3
4 3 2
4 3 2 1
```

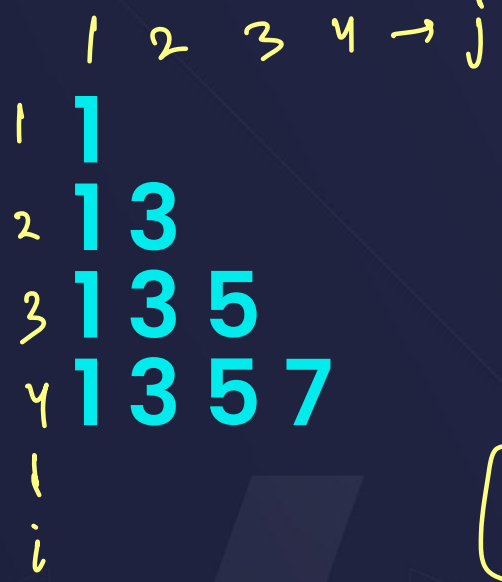
Number Triangle Ulta

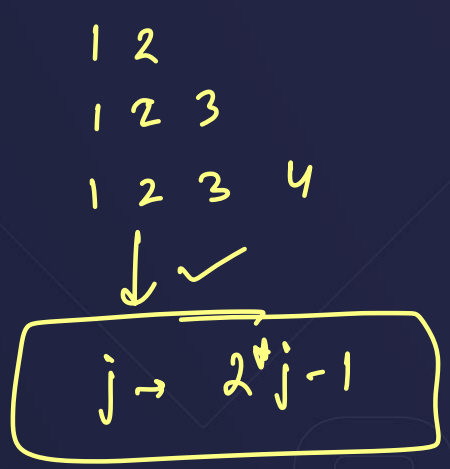
HW : Print the given pattern

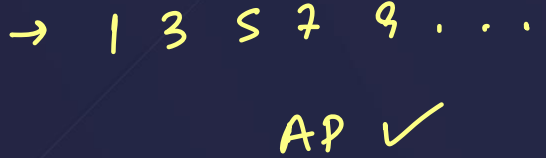
```
A B C D
A B C
A B
A
```

Alphabet Triangle Ulta

Ques : Print the given pattern

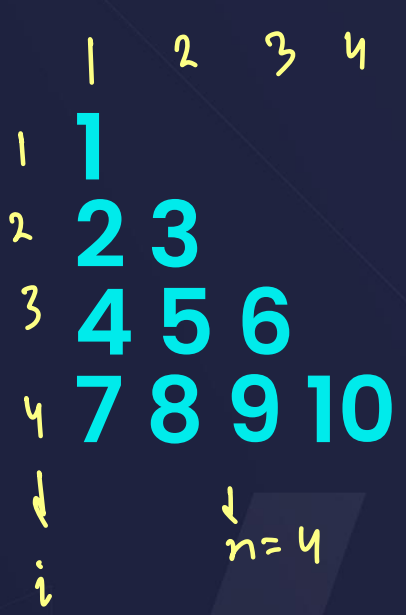






Odd Number Triangle

*Ques : Print the given pattern



n = 2

1

2 3

Floyd's Triangle

*Ques : Print the given pattern

1 2 3 4 5 → j

1 1
2 0 1
3 1 0 1
4 0 1 0 1
5 1 0 1 0 1

when
(i+j) % 2 == 0 → 1
else → 0

0 & 1 Triangle

Ques : Print the given pattern

n = odd

```
  1 2 3 4 5 → j
1 # # * # #
2 # # * # #
3 * * * * *
4 # # * # #
5 # # * # #
↓
i
```

n = 5

if (i == 3 || j == 3) → x

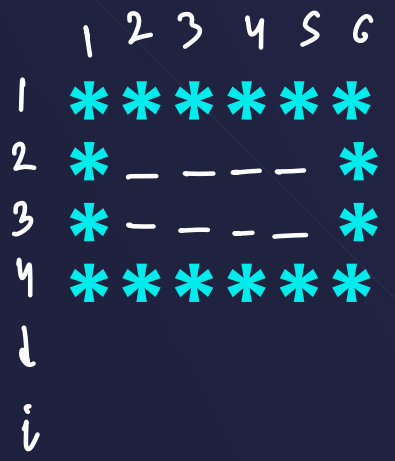
else → #

n = 3

```
# x #
x x x
# x #
```

Star Plus

HW : Print the given pattern



Stars → first row
last row
first col
last col

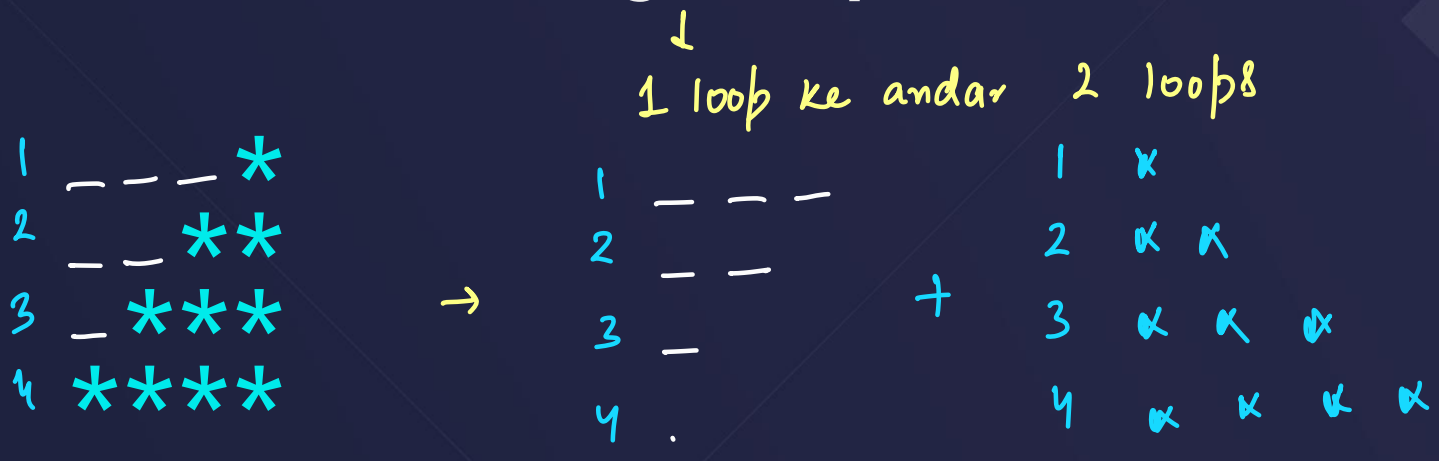
Hollow Rectangle

HW : Print the given pattern

n = odd

Star Cross

*Ques : Print the given pattern



Star Triangle Reverse

HW : Print the given pattern

```
   1
  1 2
 1 2 3
1 2 3 4
```

Number Triangle Reverse

HW : Print the given pattern

```
    A
   A B
  A B C
A B C D
```

Alphabet Triangle Reverse

HW : Print the given pattern

Rhombus

*Ques : Print the given pattern

Star Pyramid

HW : Print the given pattern

```
    1
  1 2 3
1 2 3 4 5
1 2 3 4 5 6 7
```

Number Pyramid

HW : Print the given pattern

```
    A
  A B C
A B C D E
A B C D E F G
```

Alphabet Pyramid

**Ques : Print the given pattern

Recommend : College Wallah Youtube Channel

↓

Playlists

↓

C Programming Course

↓

Lecture 4 → PP

↓

Badiya (Tough)

Star Diamond

Functions

What and Why?

DRY

↓

do not repeat yourself

↓

1) to tackle repetition

2) There are certain peice of code that we use a lot of times in problem.

Basic syntax

```
fun(){

    // code

}
```

```
void greet(){
    cout<<"Good Morning"<<endl;
    cout<<"How are you?"<<endl;
}
int main(){
    greet(); → function call
    cout<<"Hey"<<endl;
    greet(); → call
}
```

Output

- Good Morning
- How are you?
- Hey
- Good Morning
- How are you?

main() → It cannot be called more than once

return ;

↓

Keyword jisse function ka khatma ho jaata

Kind of like break;

Kaam ki baate :

1) main() ek hi baar aata hai.
2) Starts with main
3) unlimited functions

How functions work : ek ke andar doosra, doosre ke andar teesra

Parameters : formal parameters / dabbe

7 90

```
void sum(int a, int b){
    cout<<a+b;
}
int main(){
    sum(7,90);
}
```

actual parameters / values

Output

97

Pass by Value

Return type : Sum function se samajte hai

```
int sum(int a, int b){
    return a+b;
}
int main(){
    cout<<sum(72,9);
}
```

72 9 81 81 integer

return type

a, b → Sum

Output

81

Library functions

sqrt()

cbrt()

min()

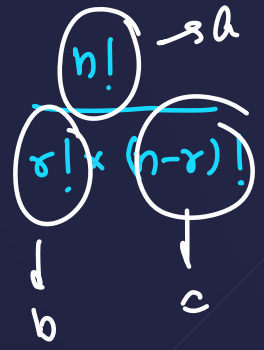
max()

pow(a,b) → a^b

Ques : Combination and Permutation

$n, r \rightarrow n \geq r$

$${}^nC_r = \frac{n!}{r! \times (n-r)!} = a/(b*c)$$



$$\hookrightarrow {}^nP_r = \frac{n!}{(n-r)!}$$

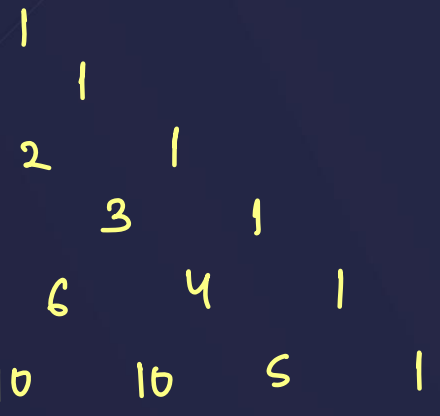
$${}^6C_4 = \frac{6!}{4! \times (6-4)!} = \frac{6!}{4! \times 2!} = \frac{\cancel{6}^{3} \times 5 \times \cancel{4} \times 3 \times \cancel{2} \times 1}{\cancel{4} \times 3 \times \cancel{2} \times 1 \times \cancel{2} \times 1} = 15$$

Ques : Pascal triangle → For Loop, Pattern Printing

Nested for Loops

	0	1	2	3	4	5 → j
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1
↓						
i						

$^{i}C_{j}$



Scope of variable

Formal parameters and Actual Parameters

variables
dabbe

Done ✓

actual values
that
are
passed

Ques : Swap 2 numbers

M-I :

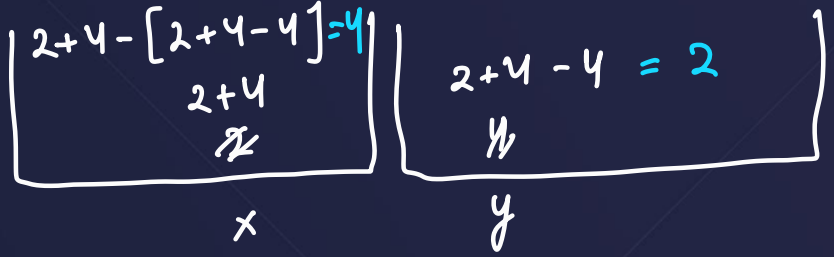
temp = x

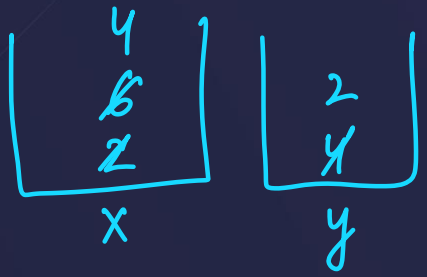
x = y

y = temp

Ques : Swap 2 numbers

M-2 : Without using extra variable





$\Rightarrow \quad x = x + y$

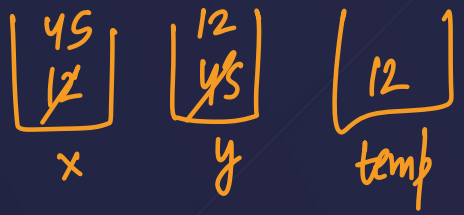
$\Rightarrow \quad y = x - y$

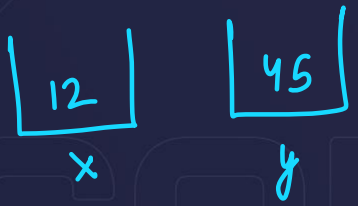
$\Rightarrow \quad x = x - y$

Pass by value & Pass by reference

```
void swap(int x, int y){
   int temp = x;
   x = y;
   y = temp;
}
int main(){
   int x = 12;
   int y = 45;
   cout<<x<<" "<<y<<endl;
   swap(x,y);
   cout<<x<<" "<<y<<endl;
}
```

Pass by Value





Output

12 45

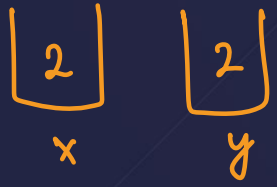
*Pointers

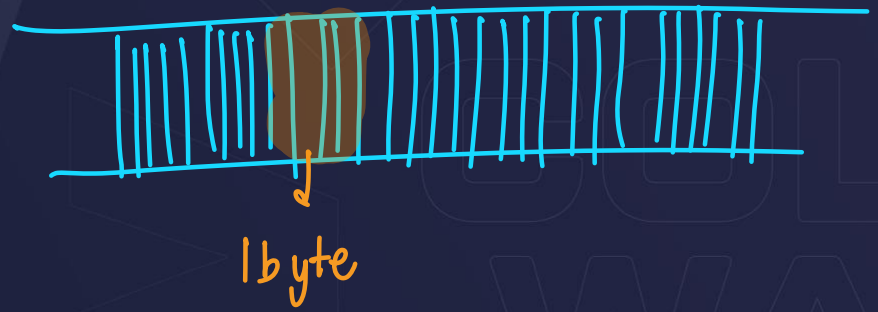
4 bytes

```
int x = 2;
```

or

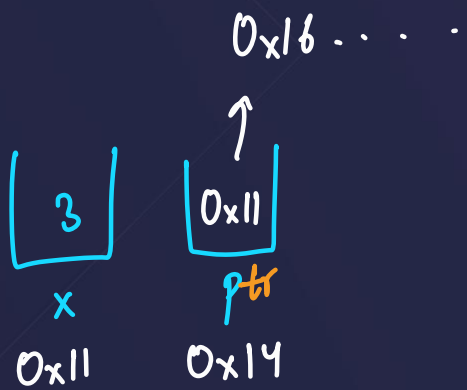
```
int x;
```





*Pointers

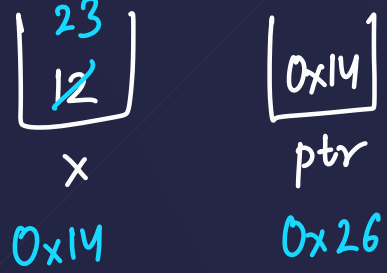
```
int x = 3;
int* ptr = &x;
cout << *ptr <<endl;
```



Access the value of of the variable whose address is in the pointer -> using dereference/* operator

*Pointers

```
int x = 12;
int* ptr = &x;
cout<<x<<endl;
*ptr = 23; // x = 23
cout<<x<<endl;
```



Output

- 12
- 23

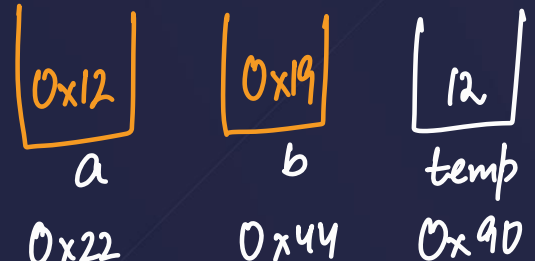
1

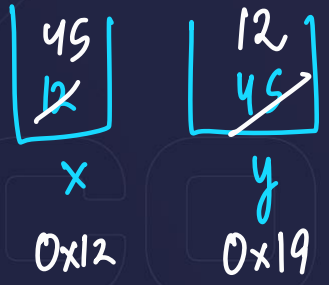
Pass by Reference

*Pointers

```
void swap(int* a, int* b){
    int temp = *a;
    *a = *b; // *a = 45
    *b = temp;
}
int main(){
    int x = 12;
    int y = 45;
    cout<<x<<" "<<y<<endl;
    swap(&x,&y);
    cout<<x<<" "<<y<<endl;
}
```

0x12 0x19





Output

12 45

Pass by Reference

State TRUE or FALSE :

1) The same variable names can be used in different functions without any conflict. YES!

State TRUE or FALSE :

2) Every called function must contain a return statement. FALSE

void

3) A function may contain more than one return statements. TRUE

State TRUE or FALSE :

5) A function may be called more than once from any other function TRUE

Recursion

Function calling itself

1) Repetition

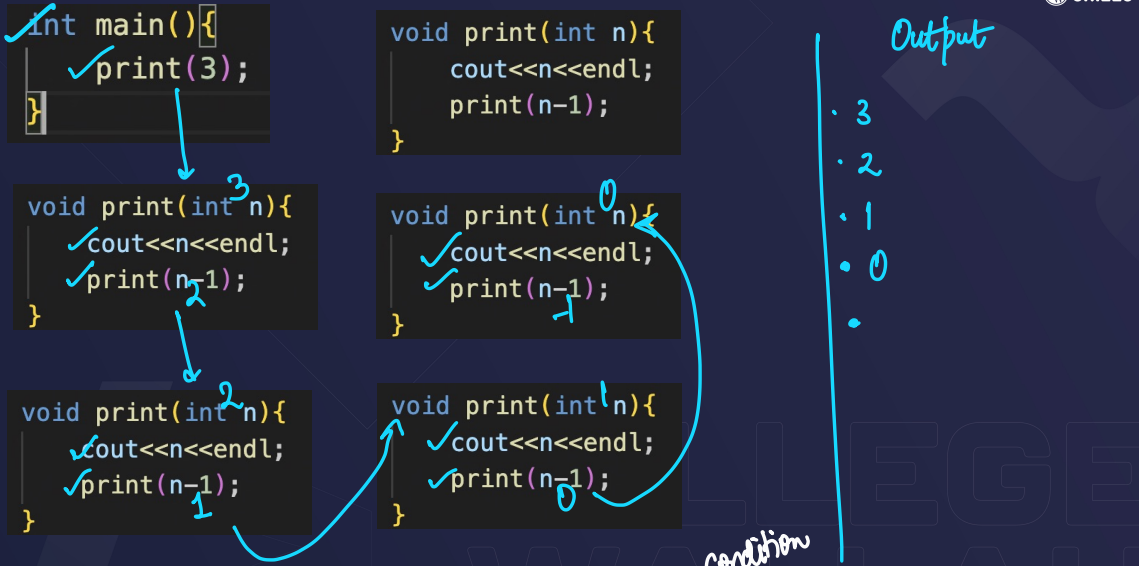
2) Infinite Loop

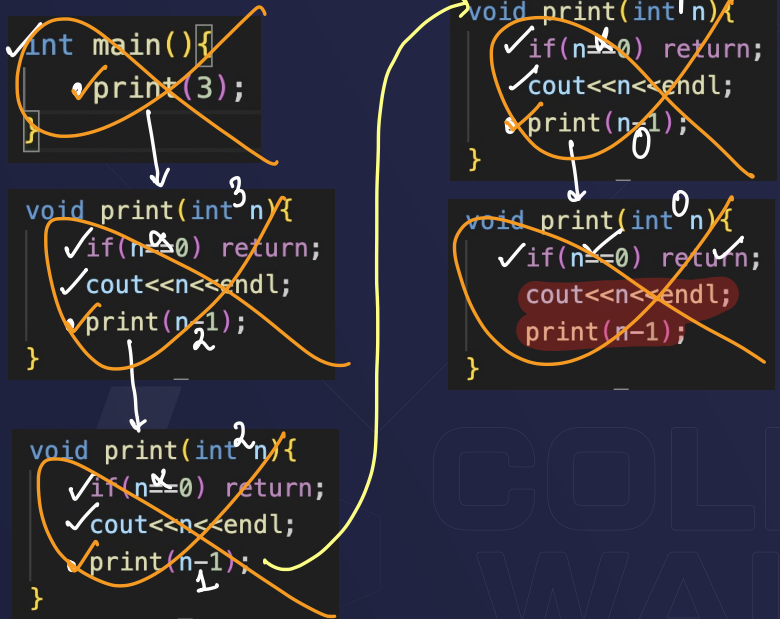
Ques : Print n to 1

print(n) → n ko print
print(n-1)

n = 5 →
5
4
3
2
1

```
void print(int n){
    cout<<n<<endl;
    print(n-1);
}
int main(){
    print(3);
}
```


int main(){
print(3);
}
void print(int n){
cout<<n<<endl;
print(n-1);
}
3
2
1
0
Output
3
2
1
0
Base Case → Condition


int main(){
print(3);
}
void print(int n){
if(n==0) return;
cout<<n<<endl;
print(n-1);
}
Output
3
2
1

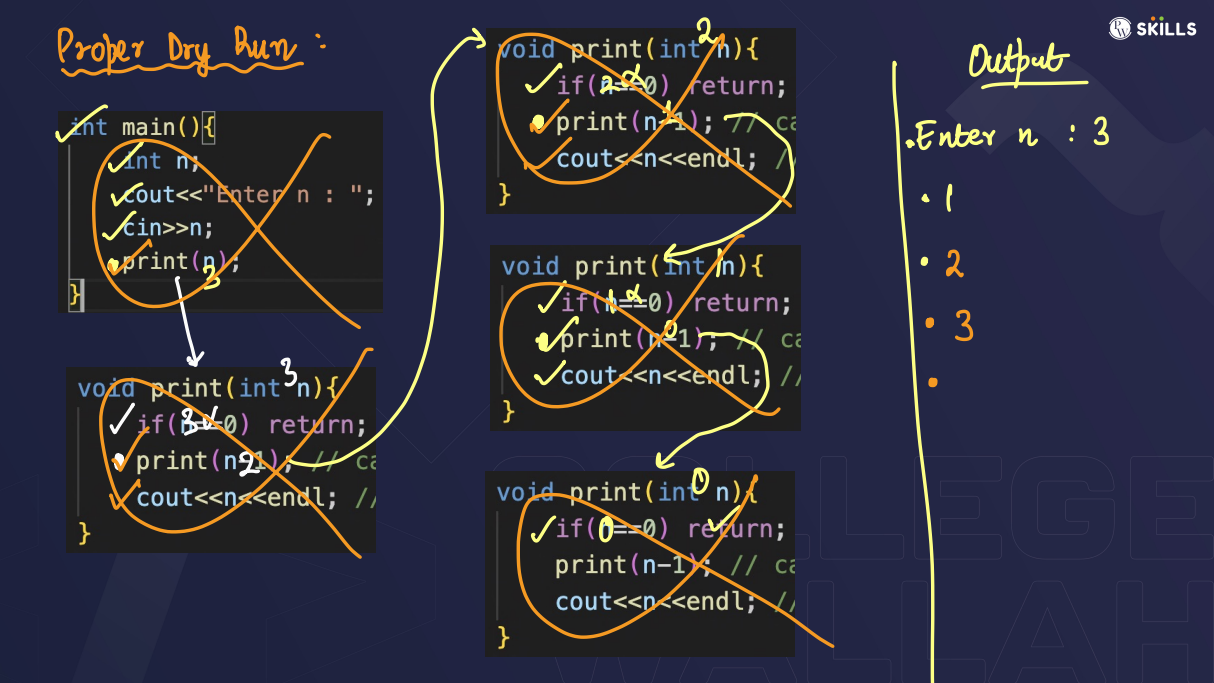
Ques : Print 1 to n [Recursion]

1
1
2
3
4
5
.
.
.

```
cout << n << endl;
print(n+1)
```

Ques : Print 1 to n (after recursive call)

↳ Without Extra Parameter


Proper Dry Run :
int main(){
int n;
cout<<"Enter n : ";
cin>>n;
print(n);
}
void print(int n){
if(n==0) return;
print(n-1); // ca
cout<<n<<endl; //
}
void print(int n){
if(n==0) return;
print(n-1); // ca
cout<<n<<endl; //
}
void print(int n){
if(n==0) return;
print(n-1); // ca
cout<<n<<endl; //
}
void print(int n){
if(n==0) return;
print(n-1); // ca
cout<<n<<endl; //
}
Output
Enter n : 3
1
2
3

Ques : Print sum from 1 to n (Parameterised)

H.W. → College Wallah

↓

Playlist

↓

C Programming

↓

Lecture → 6 Recursion

↓

Sum 1 to N parameterised

Ques : Print sum from 1 to n (Return type)

→ n = 5

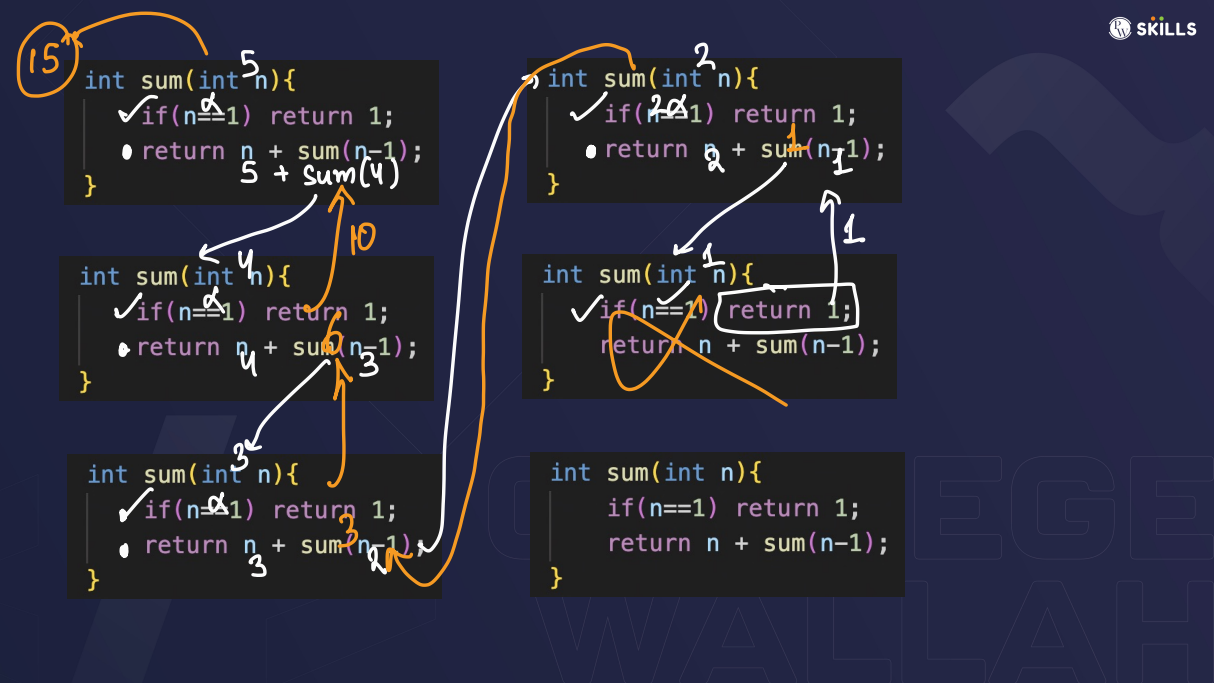
→ 1+2+3+4+5 = 15

$$S_n = \frac{n(n+1)}{2}$$

→ 1+2+3+4+5 = 5+4+3+2+1

⇒ Sum(n) = n + n-1 + n-2 + . . 3 + 2 + 1

Sum(n) = n + Sum(n-1)


15
int sum(int n){
if(n==1) return 1;
return n + sum(n-1);
}
5
5 + sum(4)
4
10
4
3
3
3
2
2
1
1
1

Ques : Make a function which calculates the factorial of n using recursion.

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

$\rightarrow n! = n \times n-1 \times n-2 \times \ldots \ldots 3 \times 2 \times 1$

$\rightarrow n! = n \times (n-1)!$

$\downarrow$

$fact(n) = n \times fact(n-1)$

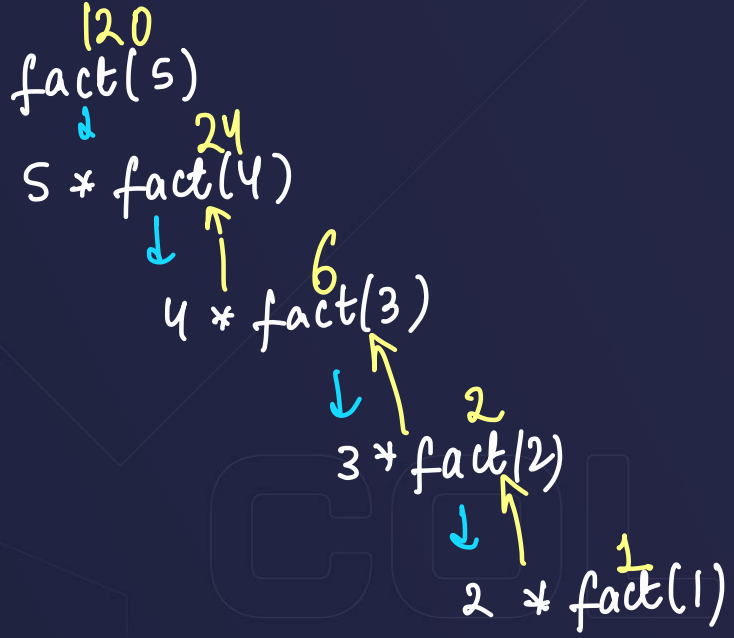
$\downarrow$

$fact(1) = 1$

Ques : Make a function which calculates the factorial of n using recursion.

Tree Diagram :

n=5



Ques : Make a function which calculates 'a' raised to the power 'b' using recursion.

$a = 3$
$b = 4$ → $3^4 = 81$

$3^4 = 3 \times 3 \times 3 \times 3$

→ $a^b = \underbrace{a \times a \times a \times a \cdots a}_{b \text{ times}}$

⇒ $a^b = a \times a^{b-1}$

⇒ $pow(a,b) = a \times pow(a, b-1)$

→ base case → $b = 0$ → $a^b = 1$

*Multiple Calls

Ques : Write a function to calculate the nth fibonacci number using recursion.

1 1 2 3 5 8 13 21 34 55 89 . . .

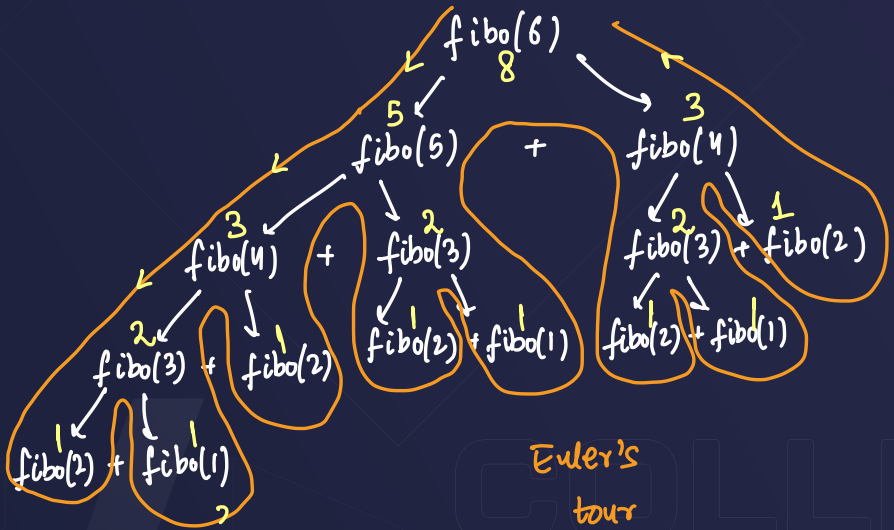
n= 1 2 3 4 5 6 7 8 9 10 11

fibo(8) = fibo(7) + fibo(6)

→ fibo(n) = fibo(n-1) + fibo(n-2);

→ fibo(1) = 1

fibo(2) = 1


fibo(6)
8
5
fibo(5)
+
3
fibo(4)
3
fibo(4)
+
2
fibo(3)
+
1
fibo(2)
2
fibo(3)
+
1
fibo(2)
2
fibo(3)
+
1
fibo(2)
+
1
fibo(1)
1
fibo(2)
+
1
fibo(1)
1
fibo(2)
+
1
fibo(1)
Euler's
tour
tree

HW : Power function (logarithmic)

$a^b = a \times a^{b-1}$

$a^b = a^{b/2} * a^{b/2}$

$2^8 = 2 \times 2^7$

$2^7 = 2 \times 2^6$

$2^6 = 2 \times 2^5$

$2^5 = 2 \times 2^4$

$2^4 = 2 \times 2^3$

$2^3 = 2 \times 2^2$

$2^2 = 2 \times 2^1$

$2^1 = 2 \times 2^0$

$2^8 = 2^4 \times 2^4$

$2^4 = 2^2 \times 2^2$

$2^2 = 2^1 \times 2^1$

$2^1 = 2 \times 2^0$

DSA → data structures & algorithms

↓

data store → variable → int x;

Arrays

What is an array?

List → list of integers → 11, 76, 13, 88, 46

list of char → A, B, A, C, F

list of float → 99.4, 93.1, 90.8, 94.6, 50.5

ek declaration se multiple dabbe bana sakta hoon.

Syntax and Declaration

Indexing

```
int x[5];
```

x[0] = 4 → initialising of array elements (individual)

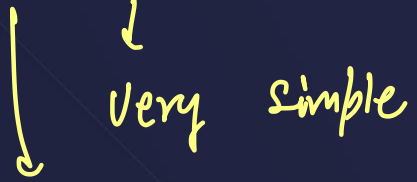
x[1] = 2

x[4] = 10

x[2] = 6

x[3] = -2

Accessing Elements of Array



very simple

using square brackets

arr[4];

Printing Output and Taking Input

Ques : Are the following array declarations correct?

int a (25) ; ✗ int a[25];

int size = 10, b[size] ; ✓

int c = {0,1,2} ; ✗ int c[3] = {0, 1, 2};

or

int c[] = {0,1,2};

int size = 10;

int b[size];

Ques : Which element of the array does this expression reference?

num[4]

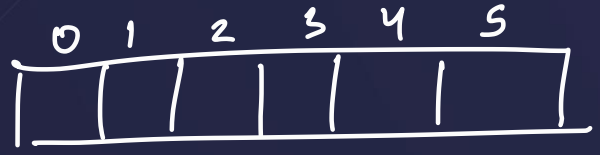
↓

5th element from start

&

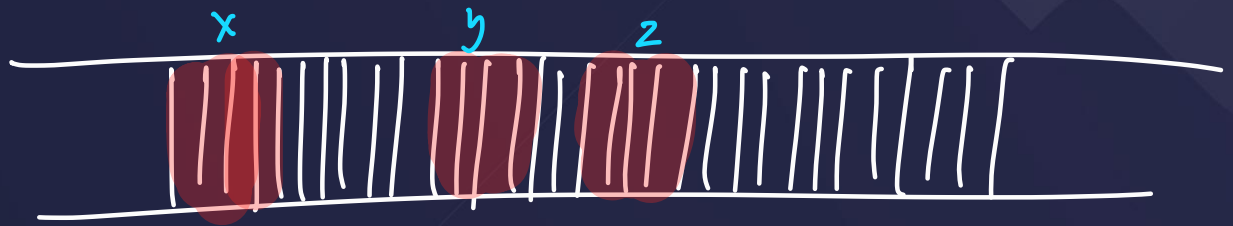
the 4th index element

int num[6];

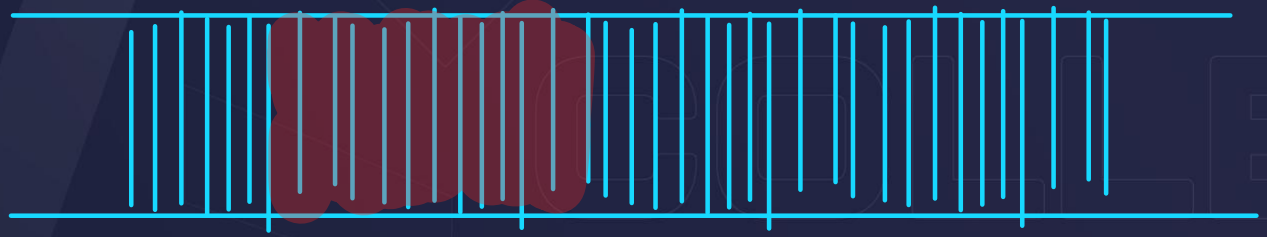


Memory Allocation in Arrays

```
int x = 4;
int y = 3;
int z = 2;
```




```
int arr[3] = {4, 3, 2 3;
```

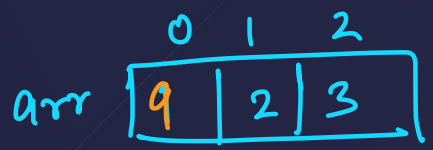


Passing Array to Functions

```
void change(int arr[]){
    arr[0] = 9;
}
int main(){
    int arr[3] = {1,2,3};
    for(int i=0;i<=2;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    change(arr);
    for(int i=0;i<=2;i++){
        cout<<arr[i]<<" ";
    }
}
```

↓

Reference



Output

1 2 3

Ques : Calculate the sum of all the elements in the given array.

```
                                  0  1  2  3  4  5
int arr[] = {5, 1, 2, 4, 6, 3};
    |
  sum = 0;
  sum += arr[0];
  sum += arr[1]
    .
    .
    .
  sum += arr[5];
```

Homework : Calculate the product of all the elements in the given array.

Ques : Find the maximum value out of all the elements in the array.

```
                0  1  2  3  4
int arr[] = { 1, 5, 6, 4, 3 };
```

```
int mx = arr[0];
```

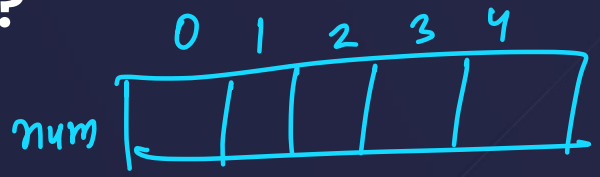
mx = 1 ⇒ 6

↓

```
mx = max(mx, arr[i])
```

Homework : Find the minimum value out of all the elements in the array.

HW : What is the difference between the 5's in these two expressions?




```
int num[5];
num[5] = 11;
```

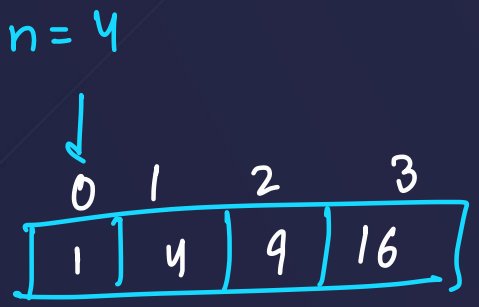
1. first is particular element, second is type
2. first is array size, second is particular element
3. first is particular element, second is array size
4. both specify array size

State TRUE or FALSE :

1. The array int num[26] has twenty-six elements. TRUE
2. The expression num[1] designates the first element in the array FALSE
3. It is necessary to initialize the array at the time of declaration. FALSE
4. The expression num[27] designates the twenty-eighth element in the array. TRUE

HW : Given an integer n. Create an array containing squares of all natural numbers till n and print the elements of the array.

```
int n;
cin>>n;

int arr[n];
```



Homework ;

Ques : Given an array of integers, change the value of all odd indexed elements to its second multiple and increment all even indexed value by 10.

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

↳ {1, 4, 3, 8, 5}

↳ {11, 4, 13, 8, 15}

Homework

Ques : Count the number of elements in given array greater than a given number x.

```
int arr[] = { 1, 3, 0, 10, 2, 5, 6 };
int x = 4;
    int count = 0;
↳   if (arr[i] > x) count++;
```

HW : Find the difference between the sum of elements at even indices to the sum of elements at odd indices.

sumEven = 9

sumOdd = 18

↳ 9 - 18 = -9

Ques : Find the second largest element in the given Array.

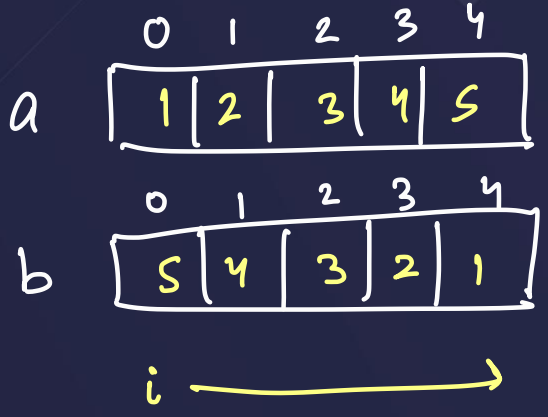
```
int arr[] = { 10, 1, 0, 9, 4, 12, 1, 2 };

int mx = INT_MIN;

for(int i=0; i<n; i++){
|   mx = max(mx,arr[i]);
}

int smax = INT_MIN;
for(int i=0; i<n ; i++){
|   if(arr[i] != mx) smax = max(smax, arr[i]);
}
```

mx = INT_MIN to 12

smax = INT_MIN 10

Ques : Write a program to copy the contents of one array into another in the reverse order.

```
int a[5] = { 1, 2, 3, 4, 5 };
int b[5];
for (i= 0; i<=4; i++ ){
    j = n-1-i;
    b[i] = a[j];
}
```



$i+j = n-1$

$j = n-1-i$

Ques : Write a program to reverse the array without using any extra array.

```
                0  1  2  3  4
int a[] = {5, 2, 3, 4, 1};
                   i
                   j
```

```
i = 0;
j = n-1;
```

Hint 1) Swap → a[i], a[j]

2) Two variables
   ↓
   i & j

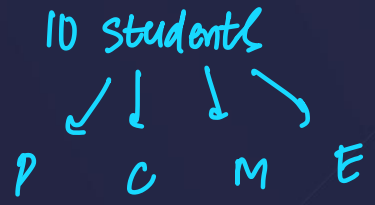
Homework : If an array arr contains n elements, then check if the given array is a palindrome or not.

arr = {1, 2, 3, 4, 3, 2, 1} ✓

= {1, 2, 3, 3, 2, 1} ✓

= {1, 2, 3, 4, 2, 1} ✗

1) Basics
2) If Else
3) Loops
4) PP
5) Function & Pointers
6) Recursion
7) Arrays
8) 2D Arrays



2D Arrays

Array → List

2D Array → Table

What and Why?

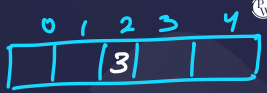
So far we have explored arrays with only one dimension. It is also possible for arrays to have two or more dimensions. The two dimensional array is also called a matrix. grid

int arr[r][c]; (c → columns, r → rows)

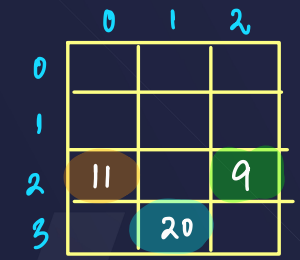
This is a 2D array where r depicts number of rows in matrix and c depicts number of columns in the matrix.

declaration :

↳ int arr[4][3];

int a[5] = 

a[2] = 3

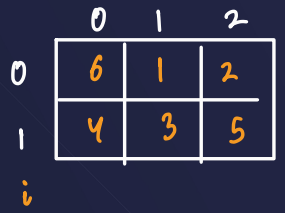


→ arr[2][2] = 9

arr[3][1] = 20

arr[2][0] = 11

int arr[2][3];



→ 6 1 2
4 3 5 ,

6 4
1 3
2 5

arr[0][0] = 6
arr[0][1] = 1
arr[0][2] = 2

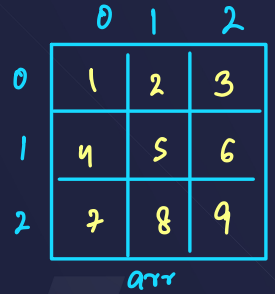
arr[1][0] = 4
arr[1][1] = 3
arr[1][2] = 5

Initialisation of a 2-Dimensional Array

```
int arr[4][2] = { { 1234, 56 }, { 1256, 43 }, { 1434, 32 }, { 1312, 96 } } ;

int arr[4][2] = { 1234, 56 , 1256, 43 , 1434, 32 , 1312, 96} ;

int arr[2][3] = { 12, 34, 56, 78, 91, 23 } ;

int arr[2][3] = {12, 34, 56, 78, 91, 23 } ;
```

2D array → array of array

```
int arr[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
```




```
int arr[][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
int arr[][3] = { 1,2,3,4,5,6,7,8,9 };
```

Ques : Write a program to store roll number and marks obtained by 4 students side by side in a matrix.

4 students → Raghav, Harsh, Sanket, Urvi

76	13	82	88
81	76	91	90

		0	1
R	0	76	81
H	1	13	76
S	2	82	91
U	3	88	90

	0	1	2	3
0	76	13	82	88
1	81	76	90	90
	R	H	S	U

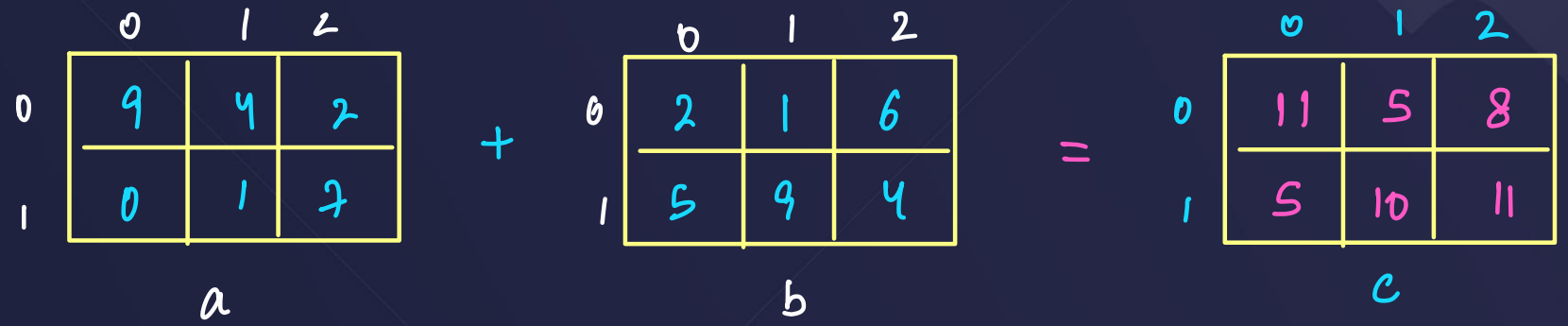
Ques : Write a program to store 10 at every index of a 2D matrix with 5 rows and 5 columns.

```
int arr[5][5];
```

	0	1	2	3	4
0	10	10	10	10	10
1	10	10	10	10	10
2	10	10	10	10	10
3	10	10	10	10	10
4	10	10	10	10	10

Homework

~~Ques~~ : Write a program to add two matrices.



→ `c[i][j] = a[i][j] + b[i][j];`

Ques : Find the maximum element in a given matrix.

↓

Ditto Same as 1D array

↓

```
int mx = INT_MIN;
```

↳ using nested loop traverse the entire 2D array

↓

```
mx = max(mx, arr[i][j]);
```

HW : Find the minimum element in a given matrix.

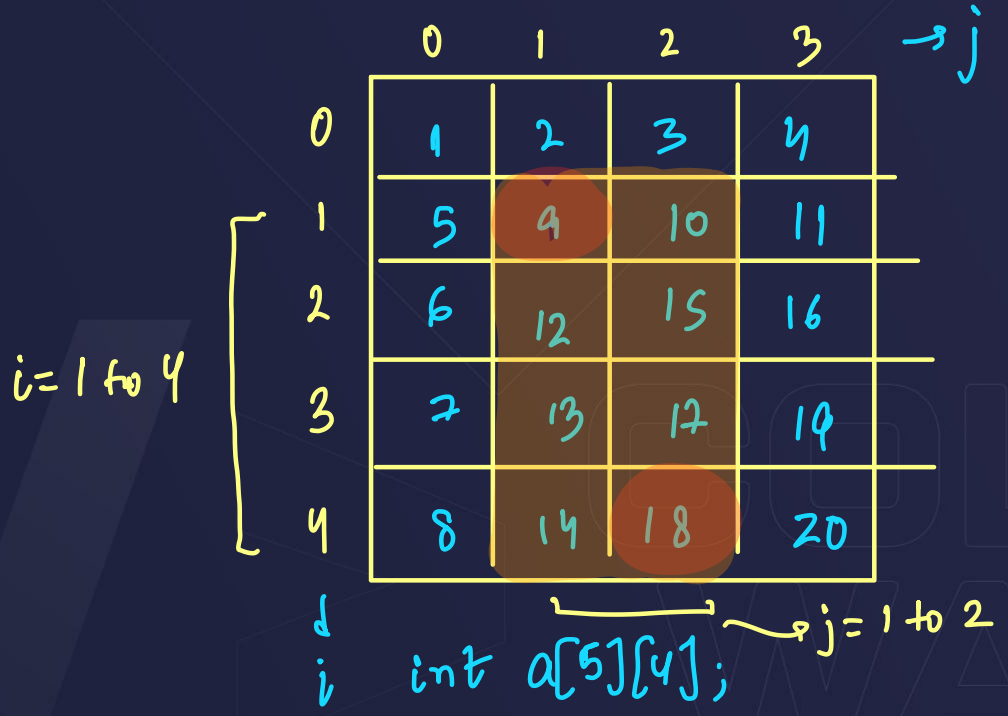
HW : Find the sum of all elements in a given matrix.

```
{
int sum = 0;
 |
sum += arr[i][j];
```

HW : Find the product of all elements in a given matrix.

```
int product = 1;

product *= arr[i][j];
```

HW : Given a matrix 'a' of dimension n x m and 2 coordinates (l1, r1) and (l2, r2). Return the sum of the rectangle from (l1,r1) to (l2, r2).



(1, 1) and (4, 2)

Ques : Write a program to Print the transpose of the matrix entered by the user.

→ H.W. Create the transpose ↓ store it in a new matrix

↓ Already done

Column Wise Printing

	0	1	2
0	1	2	3
1	4	5	6

2×3

→

	0	1
0	1	4
1	2	5
2	3	6

3×2

1) Vector, 2D vector

2) DSA

↓

Time & Space Complexity

Sorting , Searching

Linked list, Stack, Queue, Trees, Maps, Sets, Heaps

Graphs , DP , OOPS

DW Skills → Decode 2.0 Batch → C++, DSA (Paid) ← Urvi, Sanket

DW Skills → College Wallah → Playlist → C++ & DSA foundation course

1D Array

↓

integer array

float, char

raghav

↓

char array[6] =

{'r','a','g','h','a','v'}

Strings

↓

?

What are strings and Why are they used?

→ Char Arrays

Questions

string str;

Declaration of Strings and taking Input

→ string s = "Raghav Garg";

```
string s;
cin>>s;
cout<<s;
```

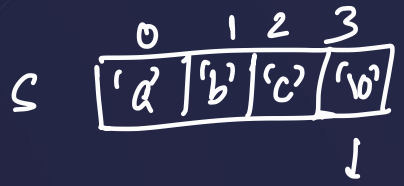
```
getline(cin,s);
```

Indexing of characters in Strings

```
String s = "raghav";

cout << s;

cout << s[0];    // r
```

length().

```
String s = "abc";
```



'\0' → null character

↓

ASCII → 0

Ques : Input a string of length n and count all the vowels in the given string.

Updation of a single character in string

H.W.

~~Ques~~ : Input a string [redacted] and Update all the even positions in the string to character 'a'. Consider 0-based indexing.

```
                    0 1 2 3 4 5 6 7 8 9 10
String s = "Raghav Garg";
```

↳ aaahavaGara

Built-in string functions

size() / length()

I

baat kar li hai

push_back() & append()

```
string s = "raghav";
```

pop_back() & clear()

↓

String ko empty string

“+” operator

reverse()

```
string s = "raghav";
```

Ques : Input a string of even length and reverse the first half of the string.

→ String s = "raghav";

i j
0 1 2 3 4 5

↳ garhav

Homework

Ques : Input a string of length greater than 5 and reverse the substring from position 2 to 5 using inbuilt functions.

```
String s = "raghav";
           012345

{

reverse( s.begin()+2 , s.begin()+5);
                                 |
                                 6
```

to_string() → int to string

```
int x = 12345;
string s = to_string(x);
```

Ques : Return the total number of digits in a number without using any loop.

Hint : Try using inbuilt to_string() function.

x = 91632

stoi

↓

```
string s = "1412361";

int x = stoi(s);
```

Maza aa gaya

Thank you!