Page No. ____
Date | 28 | 8 | 23

Conditionals & Loop :-

1. If statement -

- Syntax :-
```
if (condition)
{
    cout << "Inside if";
}
```

→ Flow of Execution :-

- The condition inside parenthesis evaluated 'true', then the code inside curly brackets will executes.
- Otherwise, it will skip the if statement block and executes the code written after that.

Ex :-
```
if (balance >= 10)
{
    cout << "Maggi";
}
```

2. If-else statement -

- syntax :-
```
if (condition)
{
    cout << "Logic";
}
else
{
    cout << "Logic 2";
}
```

Ex :-
```
if (balance >= 10)
{
    cout << "Maggi";
}
else
{
    cout << "kurkure";
}
```

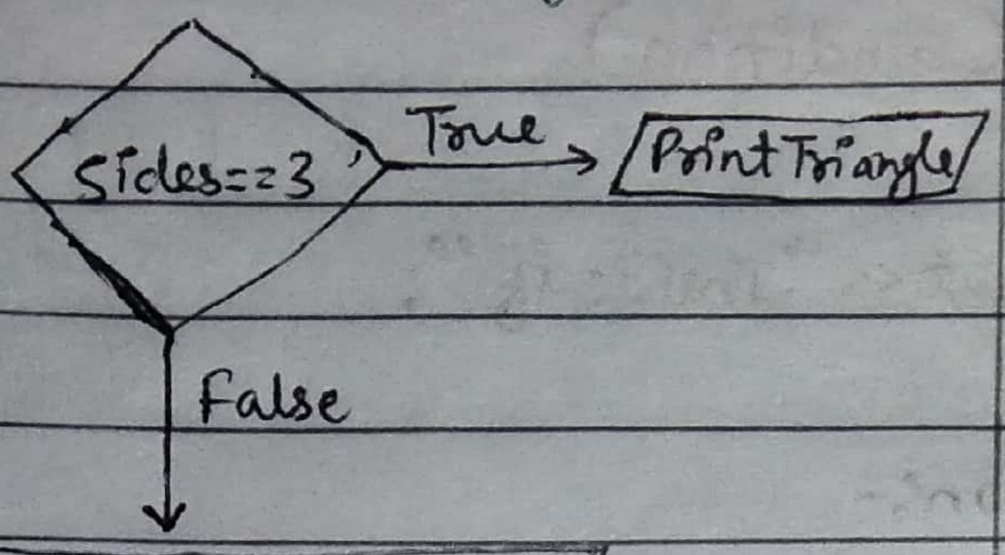
→ Flow of execution :-

- The condition inside parenthesis evaluated "true" then logic written inside curly brackets of "if" block will executes.
- Otherwise, if block will skipped and logic written inside else block will executes.

Date

Decision making block & if-else block :-

- Decision making block



- if-else block

```
if (sides==3)
{
  cout<<"Triangle";
}
else
{
  cout<<"Not a Triangle";
}
```

3. if-else if -else statement-

- Syntax:-

```
if (condition 1)
  { logic 1 }
else if (condition2)
  { logic 2 }
else if (condition3)
  { logic 3 }
else
  { logic }
```

Points to remember:

- We can use multiple else if statements.
- else block is completely optional.
- Write 'if' block only once.

→ Flow of execution:-

- If condition 1 becomes true then logic 1 will executes.
- If condition 1 fails, then it will check the other conditions of every 'else if' block and executes the logic corresponding to that block and skip the other blocks and executes the code after else.

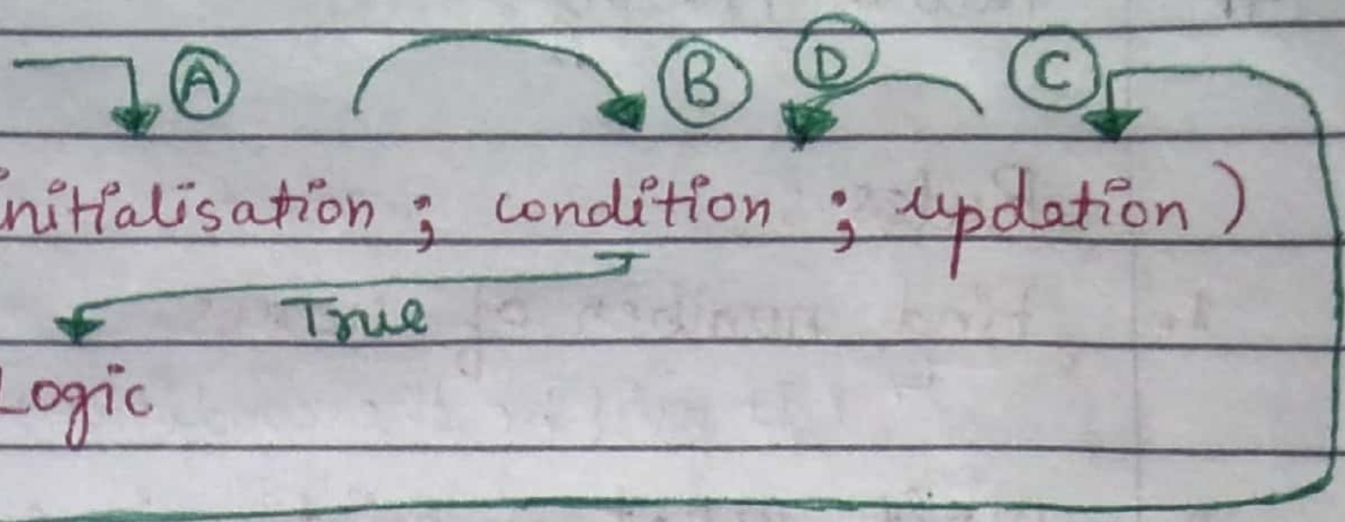
Ex:-

```
if (num==0)
  { cout<<"Its Zero"; }
else if (num>0)
  { cout<<"Its Positive"; }
else
  { cout<<"Its Negative"; }
```

Date

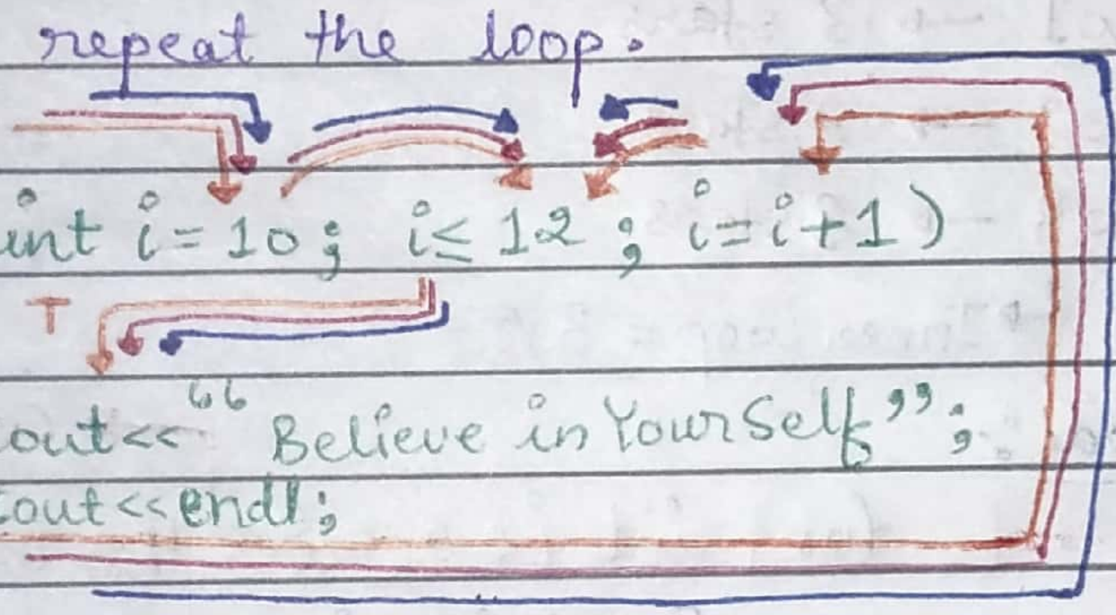
Loops :-

1. for Loop :-



→ Flow of execution :-

- Initialisation of iterator tooks place (only at once).
- If the condition evaluates 'true' then the logic written inside curly brackets will executes.
- Then iterator will gets updated and again checks the condition.
- If condition becomes false, then the loop gets terminated, otherwise it will wait to become false and repeat the loop.



Ex :- for (int i = 10 ; i <= 12 ; i = i+1)
{
cout << "Believe in Yourself";
} cout << endl;

Dry Run

	i	i <= 12	cout	i = i+1
iteration 1	i = 10	10 <= 12 (T)	Believe in Yourself	i = 10+1 = 11
iteration 2	i = 11	11 <= 12 (T)	Believe in Yourself	i = 11+1 = 12
iteration 3	i = 12	12 <= 12 (T)	Believe in Yourself	i = 12+1 = 13
iteration 4	i = 13	13 <= 12 (F) → End		

Pattern Printing :-

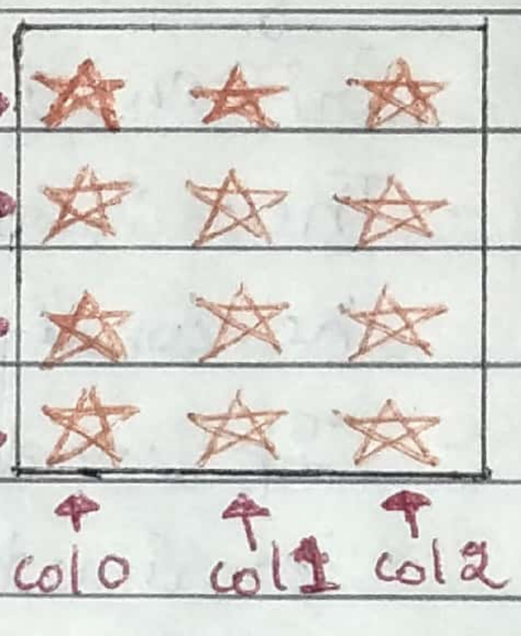
Rules :-

1. Find number of rows.
   ↳ (It will be the condition for Outer Loop.)
2. Write what is printing in each row.
   ↳ (It will be the condition for Inner Loop.)

Q1. Print rectangle pattern.

- Rule :-

1. No. of rows = 4
   ↳ Outer Loop = 4 Times
2. row 0 → 3 stars
   row 1 → 3 stars
   row 2 → 3 stars
   row 4 → 3 stars
   ↳ Inner loop = 3 ☆



code :-

```
for ( int i=0; i < 4; i=i+1) {
    for ( int j=0; j<3; j=j+1) {
        cout << " * ";
    }
    cout << endl;
}
```

→ Condition for Outer loop

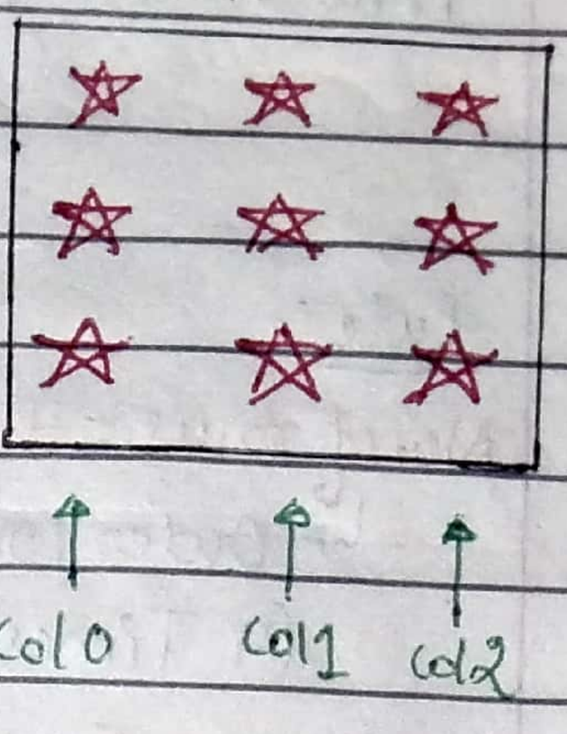
↳ Condition for Inner Loop

Flow of execution :-

- for every iteration of i from '0' to '4', j will executes itself from '0' to '3'.

Page No. _____
Date | | | |

Q2. Print square pattern.



Rule:-

1. No. of rows = 3
   ↳ Outer loop = 3 Times

2. Row 0 → 3 ☆
   Row 1 → 3 ☆
   Row 2 → 3 ☆
   ↳ Inner loop = 3 ☆

Code :-

```
for (int i=0; i<3; i=i+1)
{
    for (int j=0; j<3; j=j+1)
    {
        cout << " ☆ ";
    }
    cout << Endl;
```

Flow of execution :-

- for every single iteration of 'i', inner loop will gets executed until its condition becomes false.

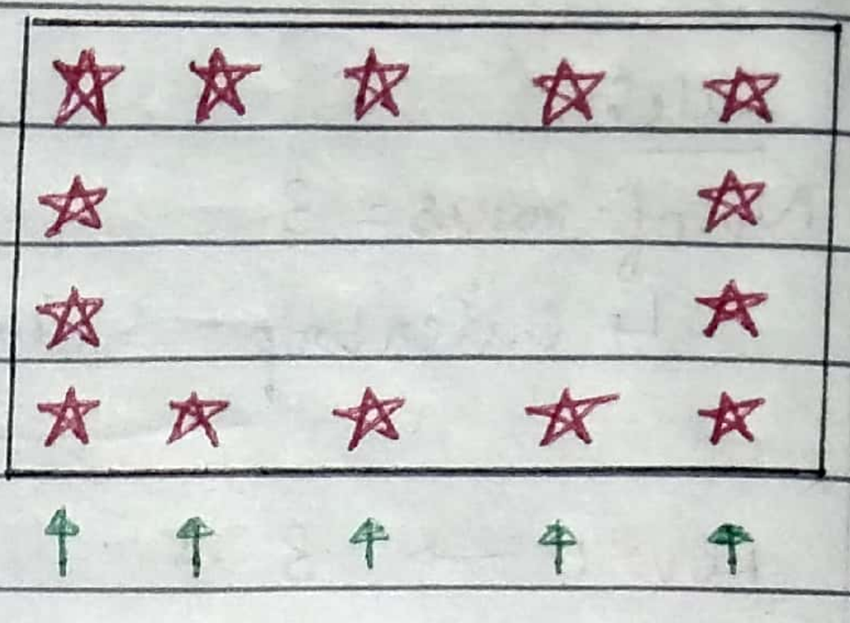
i=0 & i<3 (True)
↳ ☆ ☆ ☆ (j=0 & j<3 (True))

i=1 & i<3 (True)
↳ ☆ ☆ ☆ (j=1 & j<3 (True))

i=2 & i<3 (True)
↳ ☆ ☆ ☆ (j=2 & j<3) (True)

i=3 & i<3 (false)
↳ end

Q3. Print Hollow Rectangle.



Rule:-

1. No. of rows = 4
   ↳ Outer Loop (4 Times)

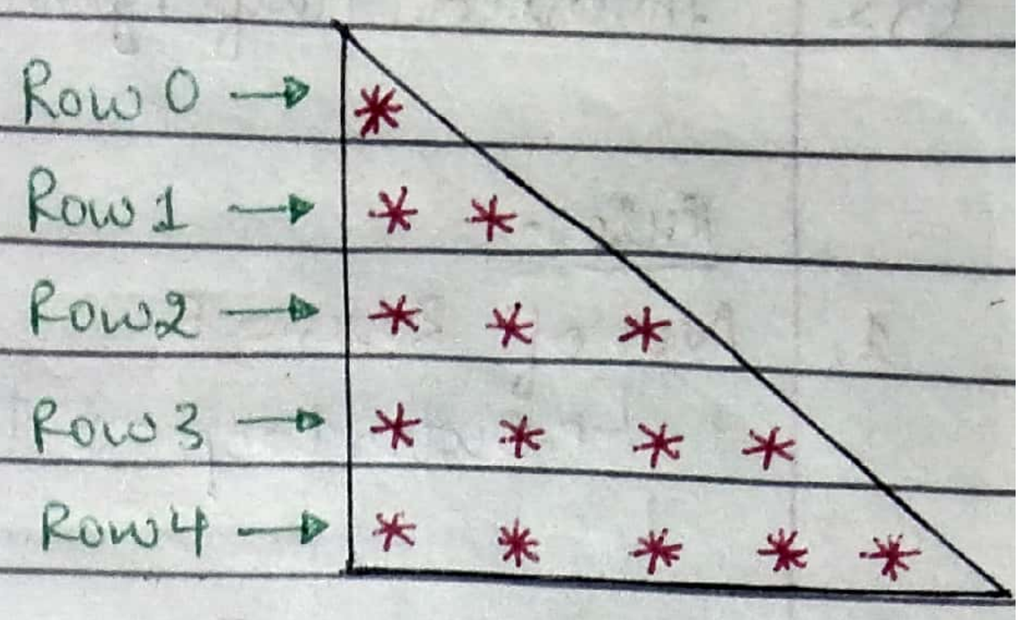
2. row 0 → 5 stars
   row 1 → 1 ★, 3 spaces, 1 ★
   row 2 → 1 ★, 3 spaces, 1 ★
   row 3 → 5 stars
   ↳ Inner Loop (5 Times)
   - if row = 0 || row 3
     ↳ 5 stars
   - else
     - if (col = 1 || col = 5)
       ↳ 1 ★
     - else
       ↳ space

Code:-

```
for(int row=0; row<4; row=row+1)
{
    for(int col=0; col<5; col=col+1){
        if(row==0 || row==3){
            cout<<"*";
        }
        else{
            if(col==0 || col==4){
                cout<<"*";
            }
            else{
                cout<<" ";
            }
        }
    }
    cout<<endl;
}
```

Page No. _____
Date | | | |

Q4. Half Pyramid Pattern.



Rule:-

1. No. of rows = 5
   ↳ Outer loop = 5 Times

2. Row 0 → 1 ★
   Row 1 → 2 ★
   Row 2 → 3 ★
   Row 3 → 4 ★
   Row 4 → 5 ★

   Inner Loop = Row+1 stars print
   ↳ Row+1 Times

Code:-

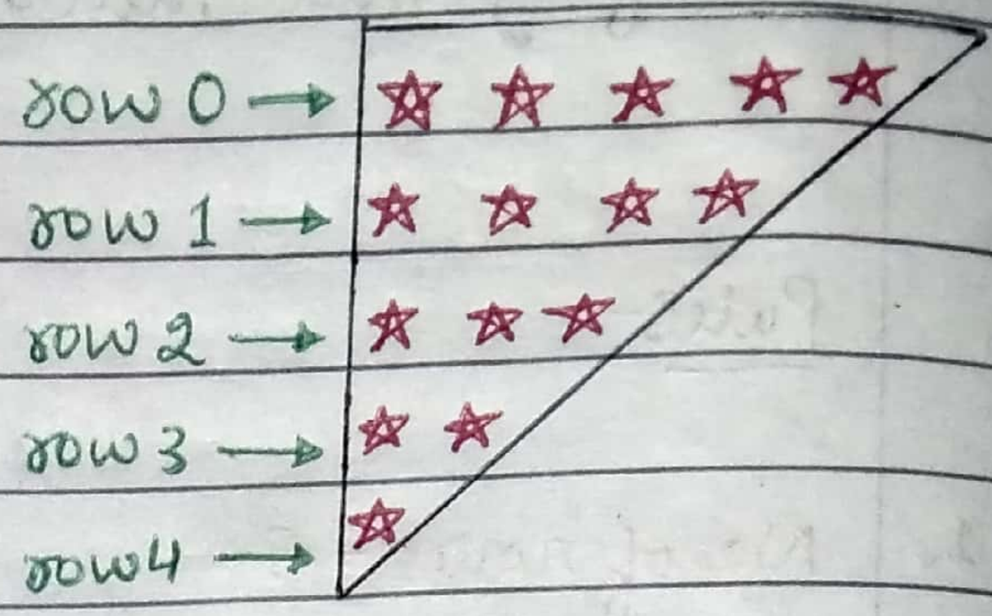
```
for (int row=0; row<5; row=row+1)
{
  for (int col=0; col<row+1; col=col+1) {
      cout << " * ";
  }
cout << endl;
}
```

Flow of execution:-

- for every iteration of 'row' from '0' to '5', it will start execution of inner loop until 'col < row+1' condition becomes false and keep printing '*'.

Page No. _____
Date

Q5. Inverted Half Pyramid.



Rule :-

1. No. of Rows = 5
   ↳ Outer Loop = 5 Times

2. row 0 → 5 ★
   row 1 → 4 ★
   row 2 → 3 ★
   row 3 → 2 ★
   row 4 → 1 ★

   Inner loop = 5 - row

Code :-

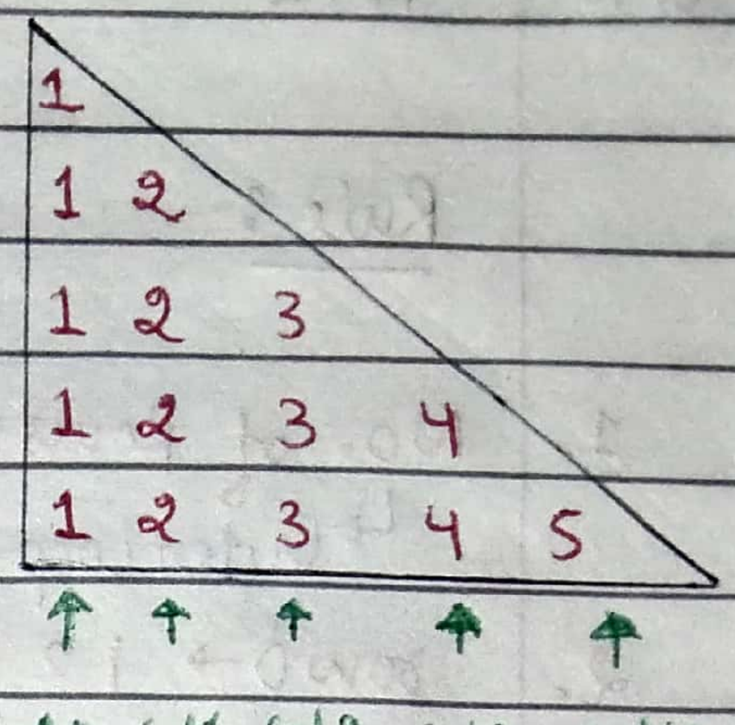
```
for ( int row=0 ; row<5 ; row=row+1)
{
    for ( int col=0; col < 5-row; col=col+1)
    {
        cout << " * ";
    }
    cout << endl;
}
```

Flow of execution :-

- for every iteration of 'row', inner loop will executes until 'col < 5-row' condition becomes false.

row = 0 & row < 5 (T) → ★ ★ ★ ★ ★ col = 0, col < 5-row (T)
row = 1 & row < 5 (T) → ★ ★ ★ ★ col = 0, col < 5-1=4 (T)
row = 2 & row < 5 (T) → ★ ★ ★ col = 0, col < 5-2=3 (T)
row = 3 & row < 5 (T) → ★ ★ col = 0, col < 5-3=2 (T)
row = 4 & row < 5 (T) → ★ col = 0, col < 5-4=1 (T)

Q6. Numeric Half Pyramid.



Rule :-

1. No. of rows = 5
   ↳ Outer Loop = 5 Times

2. row0 → 1
   row1 → 1 2
   row2 → 1 2 3
   row3 → 1 2 3 4
   row4 → 1 2 3 4 5

→ Inner loop = row + 1

What to print?

Col 0 → 1
Col 1 → 2
Col 2 → 3
Col 3 → 4
Col 4 → 5

Code :-

```
for (int row=0; row<5; row=row+1)
{
  for (int col=0; col < row+1 ; col=col+1)
  {
    cout << col+1 << " ";
  }
  cout << endl;
}
```

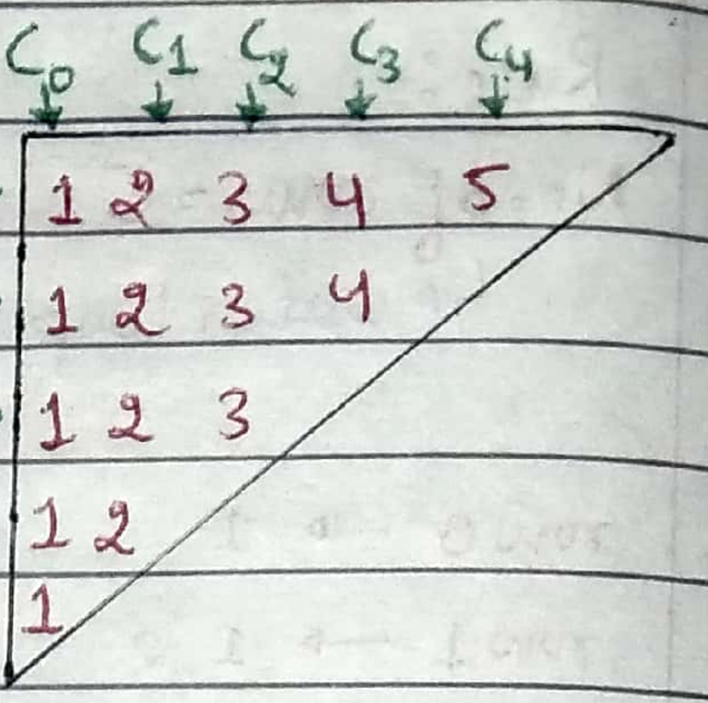
flow of execution :-

- for every row = '0' to '4', inner loop will executes until 'col < row+1' and print 'col+1' numbers.

Q7. Print Inverted Numeric Half Pyramid.

Rule :-

1. No. of Rows = 5
   ↳ Outer Loop = 5 Times
2. row 0 → 1 2 3 4 5
   row 1 → 1 2 3 4
   row 2 → 1 2 3
   row 3 → 1 2
   row 4 → 1



→ What to print

$C_0 \rightarrow 1$

$C_1 \rightarrow 2$ = col+1

$C_2 \rightarrow 3$

$C_3 \rightarrow 4$

$C_4 \rightarrow 5$

Code :-

```
for (int row=0; row<5; row=row+1)
{
   for (int col=0; col<5-row; col=col+1)
   {
      cout << col+1;
   }
   cout << endl;
}
```

flow of execution :-

- for every row = '0' to '5', inner loop will executes until 'col < 5 - row' becomes false and keep printing 'col+1'.

			C_0	C_1	C_2	C_3	C_4
row=0 & 0<5 (T) →	col=0 & 0<5-0 (T) →		1	2	3	4	5
row=1 & 1<5 (T) →	col=0 & 0<5-1 (T) →		1	2	3	4	
row=2 & 2<5 (T) →	col=0 & 0<5-2 (T) →		1	2	3		
row=3 & 3<5 (T) →	col=0 & 0<5-3 (T) →		1	2			
row=4 & 4<5 (T) →	col=0 & 0<5-4 (T) →		1				
row=5 & 5<5 (F)							

↳ end