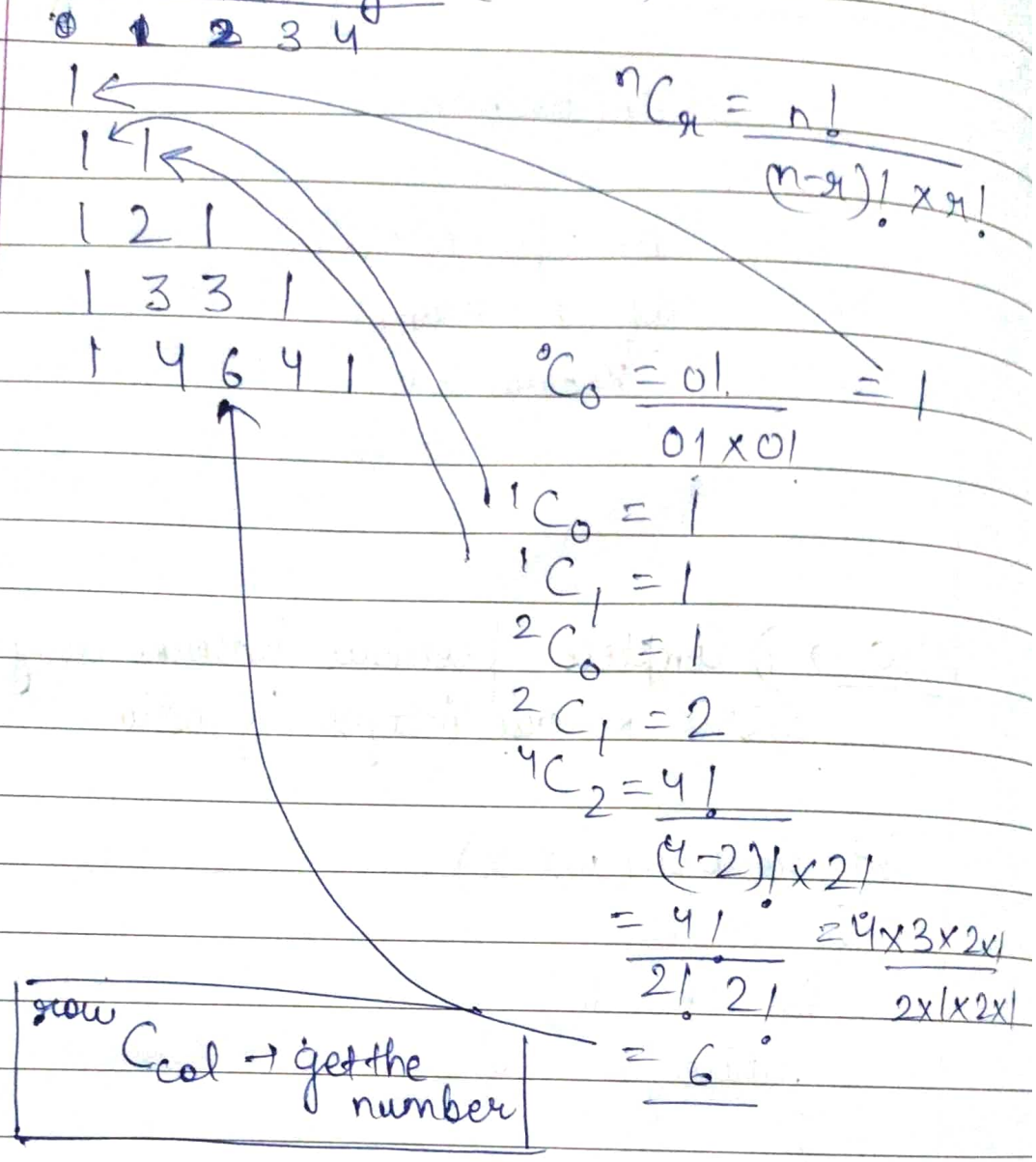
Pascal Triangle (Pattern)




```
int nCr (int n, int r)
{
    ____
    ____
}
```

Limitation of a factorial method -
Big No. ke liye factorial nhi laa paega.

```
int factorial (int n)
{
    ____
    ____
}.
```

```
int factorial (int n)
{
 int ans=1;
 for(i=n; i>=1; i--)
 {
  ans = ans * i;
 }
return ans;
}
```

```
int nCr (int n, int r)
 {
  int val1= fact(n);
  int val2 = fact(n-r);
  int val3 = fact (r);
  int ans = (val1)
            (val2 * val3)
  return ans;
 }
```

Handle integer overflow here

Butterfly Pattern

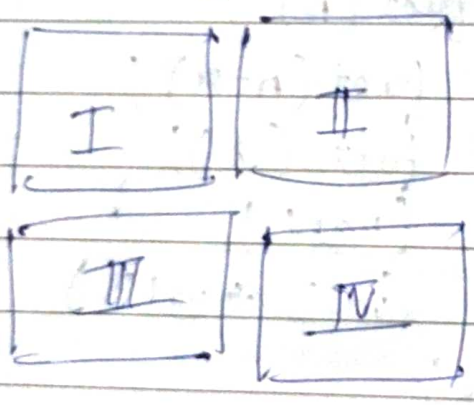
n = 3

```
*  -  -  -  *
*  *  -  *  *
*  *  *  *  *
*  *  *  *  *
*  *  -  *  *
*  -  -  -  *
```

		star	space
* _ _	I st R	1	2
* * _	II nd R	2	1
* * *	III rd R	3	0
		(row)	(n-row)

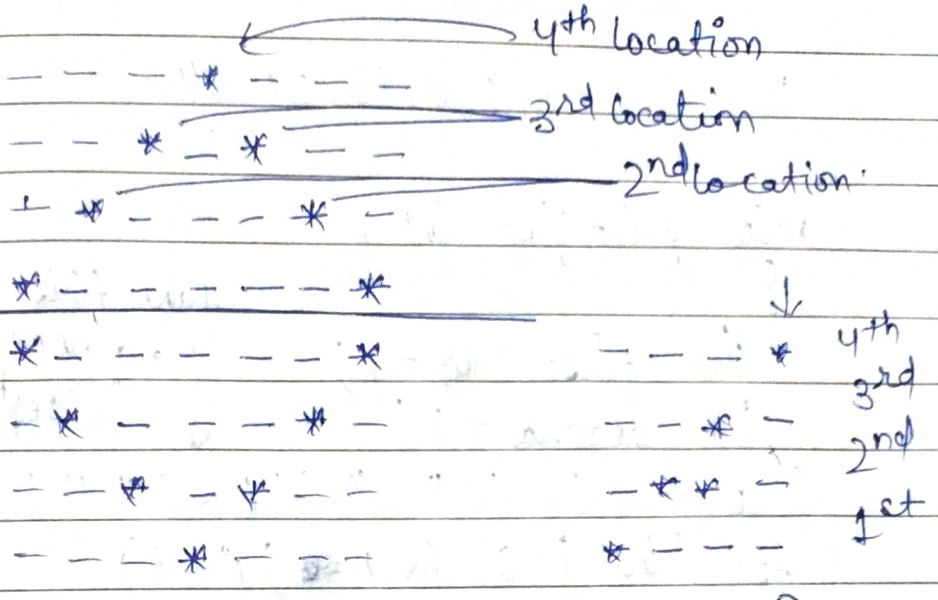
for this → for (int row → 1 <= n)
{
star →
space →
}

Break butterfly pattern into four parts-



for (int row → 1 <= n)
{
star →
space →
star →
}
cout << endl
for (row → 1 → <= n)
{
star →
space →
star →
}

Hollow diamond pattern-



I^st part

```
for (int i=1; i<=4; i++)
{
    //each row
    for(int col=1; col<=4; col++)
    {
        if (col==n-row+1)
            cout<<"*";
        else
            cout<<" ";
    }
}
```

II part

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

Page No.:
Date:

Fibonacci Series -

0, 1, 1, 2, 3, 5, 8, 13, 21, 34 ....

```
void printfibo(int n)
 {
  int first = -1;
  int second = 1;
//cout <<a<< " " <<b<< " ";
 for (int i=0; i< n; i++)
    { int sum=a+b;
      cout << sum << " ";
      a=b;
      b= sum;
    }
  }

int main()
 {
  int n;
 cout <<" Enter n" <<endl;
 cin >>n;
 printfibo(n);
  return 0;
    }
```

take first as -1 then no need to print a & b seperately