qus 20- https://code4coding.com/c-program-count-wordcharacters-and-space-of-a-string/

```
#include<iostream>
#include<string.h>

using namespace std;

int main() {
char str[200];
gets(str);
int word=1,ch=0,len=0,sp=0;
for(int i=0;str[i]!='\0';i++)
{
   if(str[i]==' ')
   {
      word++;
      sp++;
   }
   len++;

}
cout<<"Word = "<<word<<endl<<"character = "<<len-sp<<endl<<"special char = "<<sp<<endl;
            return 0;
}
```

qus 22- https://www.geeksforgeeks.org/program-to-print-solid-and-hollow-rhombus-patterns/

```
#include<iostream>
#include<string.h>

using namespace std;

int main() {
   int n;
```

```
cin>>n;
cout<<"Solid Rhombus:\n ";
for(int i=1;i<=n;i++)
{
    for(int j=i;j<n;j++)
    cout<<" ";
    for(int k=1;k<=n;k++)
    {
        cout<<"*";
    }cout<<endl;
}
cout<<"\nHollow Rhombus: \n";
for(int i=1;i<=n;i++)
{
    for(int j=i;j<n;j++)
    cout<<" ";
    if(i==1||i==n)
    {
        for(int k=1;k<=n;k++)
        cout<<"*";

    }
    else{
        for(int m=1;m<=n;m++)
        {
            if(m==1||m==n)
            cout<<"*";
            else
            cout<<" ";
        }
    }cout<<endl;
    }
    return 0;
}
```

qus no-23

Take as input N, a number. Print the pattern as given in the input and output section.

Input Format

Enter value of N

Constraints

1 <= N < 10

Output Format

Print the pattern.

Sample Input

```
7
```

Sample Output

```
 1******
 12*****
 123****
 1234***
 12345**
 123456*
 1234567
```

Explanation

There is no space between any two numbers. Catch the pattern for corresponding input and print them accordingly.

```
#include <iostream>
using namespace std;
```

```
int main() {
        int n;
        cin>>n;
        for(int i=1;i<=n;i++)
        {int j=1;
          for(j=1;j<=i;j++)
          {
            cout<<j;
          }
          for(int k=j;k<=n;k++)
          {
            cout<<"*";
          }
          cout<<endl;
        }
        return 0;
}
```

qus 24- https://www.cpp.thiyagaraaj.com/c-programs/c-pattern-example-programs/simple-c-program-for-print-inverted-triangle-pattern

```
vector<string> invIsoTriangle(int N) {
    // code here
    vector<string>v;
    for(int i=N;i>=1;i--)
    {
      string s="";
      for(int k=N-i;k>0;k--)
      s+=' ';
      for(int j=1;j<=i;j++)
      s+='*';
      for(int l=1;l<i;l++)
```

```
            s+='*';

            v.push_back(s);

        }return v;

    }
```

qus 25- Take as input N, a number. Print the pattern as given in output section for corresponding input.

Input Format

Enter value of N

Constraints

Output Format

All numbers and stars are Space separated

Sample Input

5

Sample Output

```
1 2 3 4 5

1 2 3 4 *

1 2 3 * * *

1 2 * * * * *

1 * * * * * * *
```

Explanation

Catch the pattern for the corresponding input and print them accordingly.

```
#include<iostream>

#include<string.h>


using namespace std;


int main() {

    int n;
    cin>>n;

    for(int i=n;i>=1;i--)

    {

        for(int j=1;j<=i;j++)

        cout<<j<<" ";

        int k=2*(n-i)-1;

        for(int m=1;m<=k;m++)

        {

            cout<<"* ";

        }

        cout<<endl;


    }

    return 0;

}
```

qus 26- Take N (number of rows), print the following pattern (for N = 4)

```
0

1 1

2 3 5

8 13 21 34
```

Input Format

Constraints

0 < N < 100

Output Format

Sample Input

4

Sample Output

```
0

1    1
```

```
2   3   5
8  13   21   34
```

Explanation

Each number is separated from other by a tab. For given input n, You need to print n(n+1)/2 fibonacci numbers. Kth row contains , next k fibonacci numbers.

```
#include<iostream>
#include<string.h>

using namespace std;

int main() {
   int n;
   cin>>n;
   int sum=0,num1=1,num2=1;
  for(int i=1;i<=n;i++)
  {
     if(i==1)
     cout<<0;
     else if(i==2)
     cout<<1<<"\t"<<1;
     else{
        for(int j=1;j<=i;j++)
        {
        sum=num1+num2;
        num1=num2;
        num2=sum;
        cout<<sum<<"\t";}
     }
     cout<<endl;
  }
   return 0;
}
```

27 Bubble Sort Easy

```
void bubbleSort(int arr[],int n)
{
  int i,j,temp;
  for(i=0;i<n-1;i++)
  {
    for(j=0;j<n-i-1;j++)
    {
      if(arr[j]>arr[j+1])
      {
        temp=arr[j];
      arr[j]=arr[j+1];
      arr[j+1]=temp;
      }

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

28 Selection Sort Easy

```
void selectionSort(int arr[],int n)
{
  int i,j,temp;
  for(i=0;i<n;i++)
  {
    int min=i;
    for(int j=i+1;j<n;j++)
    {
      if(arr[min]>arr[j])
      min=j;
    }
    int temp=arr[min];
    arr[min]=arr[i];
    arr[i]=temp;

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

29 Print Numbers in Reverse Order Easy

```
int reverseNumber(int n)
{
    int res=0;
    while(n)
    {
        res=res*10+n%10;
        n/=10;
    }return res;

}
```

30 Linear Search Easy

```
int linearSearch(int arr[],int n,int data)
{
 for(int i=0;i<n;i++)
 {
     if(arr[i]==data)
     return 1;

 }
 return 0;
}
```