Create a custom exception class named StackException. The Push()and Pop() method should throw object of StackException when the stack is full or empty respectively.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace exception
{
   internal class Stack
   {
      private int[] ele;
      private int top;
      private int max;
      public Stack(int size)
      {
         ele = new int[size];
         top = -1;
         max = size;
      }
      public void Push(int item)
      {
         if(top == max -1)
         {
            throw new Exception("Stavk overflow not perform push");

         }
         else
         {
            ele[++top] = item;

         }
      }
      public int Pop()
      {
         if(top==-1)
         {
            throw new Exception("stack is empty");

         }
         else
         {
            Console.WriteLine("pop element is:" + ele[top]);
            return ele[top--];
         }
      }
      public void printStack()
      {
```

```
            if(top== -1)
            {
               Console.WriteLine("stack is empty");
               return;
            }
            else
            {
               for(int i=0;i<=top;i++)
               {
                  Console.WriteLine("Item[" + (i + 1) + "]:"+ele);

               }
            }
         }
      }
      class Program
      {
         public static void Main()
         {
            Stack S = new Stack(5);
            S.Push(10);
            S.Push(20);
            S.Push(30);
            S.Push(40);
            S.Push(50);
            //S.Push(60);

            Console.WriteLine("item are:");
            S.printStack();
            S.Pop();
            S.Pop();
            S.Pop();
            S.Pop();
            S.Pop();
            Console.ReadKey();

         }
      }
}
```

Output:

```
pop element is:50
pop element is:40
pop element is:30
pop element is:20
pop element is:10
```