**Task No. 1:** Write a program to build your own stack class. The minimum your stack class should include is using your enrollment no :

A Push(Object) method

A Pop() method

A Peek() method

A IsFull() method

A IsEmpty() method

A Display() method

A Count() method

**Solution:**

using System;

namespace LAB\_5\_Stack

{

internal class Stack

{

static readonly int MAX = 10;

int top;

int[] stack = new int[MAX];

public Stack()

{ top = -1; }

public bool isempty()

{

return top < 0 ? true : false;

}

public void push(int data)

{

if ((top) >= (MAX - 1))

{

Console.WriteLine("Stack OverFlow");

}

else

{

stack[++top] = data;

}}

public int pop()

{

if (!isempty())

{

int value = stack[top--];

return value;

}

else

{

Console.WriteLine("Stack underflow");

return 0;

}}

public void peak()

{

if (!isempty())

{

Console.WriteLine("The Top Most Element Of Stack Is {0} ", stack[top]);

}

else

{

Console.WriteLine("Stack UnderFlow");

}}

public void printstack()

{

if (!isempty())

{

Console.WriteLine("Items In The Stack Are :");

for (int i = top; i >= 0; i--)

{

Console.WriteLine(stack[i]);

}

}

else

{

Console.WriteLine("Can't Print Stack Because Stack Is UnderFlow");

}}

public int count()

{

int count = 0;

if (!isempty())

{

for (int i = top; i >= 0; i--)

{

++count;

}

return count;

}

else

{

Console.WriteLine("Can't Print Stack Because Stack Is UnderFlow");

return count;

}}

public void Isfull()

{

int count = 0;

if (!isempty())

{

for (int i = top; i >= 0; i--)

{

++count;

}

if (count < MAX)

{

int remaining = MAX - count;

Console.WriteLine("The Remaining Element To Fill The Stack Is : {0}", remaining);

}

else if (count == MAX)

{

Console.WriteLine("The Stack Is Full!!!!!");

}

}

else

{

Console.WriteLine("The Stack Is EMPTY !!!!!");

}}

static void Main(string[] args)

{

Stack stack = new Stack();

byte res = 0;

do

{

Console.WriteLine("--------------------------------------------------------------------------");

Console.WriteLine("Please Choose An Option Below: ");

Console.WriteLine(" 1) Push Value");

Console.WriteLine(" 2) Pop Value");

Console.WriteLine(" 3) Peak Value");

Console.WriteLine(" 4) Check Stack IsFull() ");

Console.WriteLine(" 5) Check Stack IsEmpty() ");

Console.WriteLine(" 6) Print Stack Value");

Console.WriteLine(" 7) Count Stack Value");

Console.Write("Enter : ");

res = byte.Parse(Console.ReadLine());

if (res == 1)

{

Console.Write("Please Enter Vaue You Want To Push : ");

stack.push(int.Parse(Console.ReadLine()));}

else if (res == 2)

{

Console.WriteLine("The Pop Element is {0}", stack.pop());}

else if (res == 3)

{

stack.peak();}

else if (res == 4)

{

stack.Isfull();}

else if (res == 5)

{

if (stack.isempty())

{

Console.WriteLine("The Stack Is Empty !!!! ");

}

else

{

Console.WriteLine("The Stack Is NOT Empty !!!! ");

}

}

else if (res == 6)

{

stack.printstack();}

else if (res == 7)

{

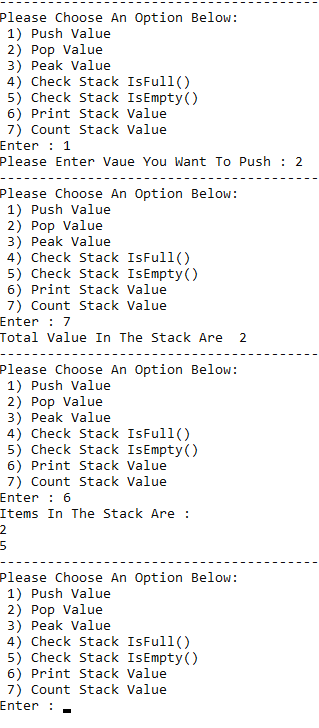
Console.WriteLine("Total Value In The Stack Are {0} ", stack.count());}

} while (res != 1 || res != 2 || res != 3 || res != 4 || res != 5 || res != 6 || res != 7);

}

}

}

Text, letter

Description automatically generated**Output:**

Text

Description automatically generated