

MID-ASSINGMENT

MOST.AFROJA MAHOMUDA

ID;20-43554-1 SEC; K

Mid- Assignment

Most Afroja Mahomuda

ID: 20-43554-1

1. Answer to the question No: 1

Data structure is a way of collecting and organising in such a way that we can perform operations on these data in effective way. In simple language, Data structures are structures programmed to store ordered data so that various operations can be performed on it easily. There are primitive Data structures and abstract Data structure. To solve the given problem, stack data-structure will be use

Stack :

Stack is an abstract data type with a bounded capacity. It is a simple data structure that allows adding and removing elements in a particular order. Everytime an element is added, it goes on the top of the stack and the only element that can removed is the element that is at the top of the stack, just like a pile of objects. The simplest application of a stack is to reverse a word.

Q. For stack data structure, two principal operations will be used which is push and pop.

Stack can be easily implemented using an array.

There is push operation for checking if the stack is full or not. If the stack is full then print error of overflow and exit the program.

If the stack is not full, then the increment the top and add the element.

There is pop operation for checking if the stack is empty or not. If the stack is empty, then print error of underflow and exit the program. If the stack is not empty then print the element at the top and decrement the top.

To solve this problem, for push and pop must need to check first that stack is full or empty. If stack shows empty, there is no way to using pop operation. On the other hand if a stack is full, push operation is restricted also. Operating push and pop on the initially empty stack and confirming that the operation is valid or not for instruction. However, pop out something is possible when element exists in the stack. pop operation is never possible in the form of an empty stack.

```
3. #include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int t,n;
```

```
    cin>>t;
```

```
    while(t--)
```

```
    {
```

```
        cin>>n;
```

```
        int ar[n],flag=0,oneval=0,zeroval=0,val,i;
```

```
        for(i=0;i<n;i++)
```

```
        {
```

```
            cin>>val;
```

```
    if(val)
    {
        oneval++;
    }
    else
    {
        zeroval++;
    }
    if(zeroval>oneval)
    {
        flag=1;
    }
}
if(flag)
{
    cout<<"Invalid"<<endl;
}
else
{
    cout<<"Valid"<<endl;
}
}
return 0;
}
```

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4
5     int t,n;
6     cin>>t;
7     while(t-->0)
8     {
9         cin>>n;
10        int ar[n],flag=0,oneval=0,zeroval=0,val,i;
11        for(i=0;i<n;i++)
12        {
13            cin>>val;
14            if(val)
15            {
16                oneval++;
17            }
18            else
19            {
20                zeroval++;
21            }
22            if(zeroval>oneval)
23            {
24                flag=1;
25            }
26        }
27        if(flag)
28        {
29            cout<<"Invalid"<<endl;
30        }
31        else
32        {
33            cout<<"Valid"<<endl;
34        }
35    }
36    return 0;
37 }
```

Process returned 0 (0x0) execution time : 93.111 s
Press any key to continue.

3.

4.

4.

For testcase 1,

this test line can operate five times and showing the confirmation output is valid. At first of all doing push operation two times on given empty stack. Then operate pop operation two times also. By doing two times push and then two times pop operation the stack gets back initial position that means the given stack is empty again. Lastly operating one more push operation and this is a valid operation, because push operation can be happen on an empty stack without any problem. That's why the test case 1 is acceptable and shows the output valid.

