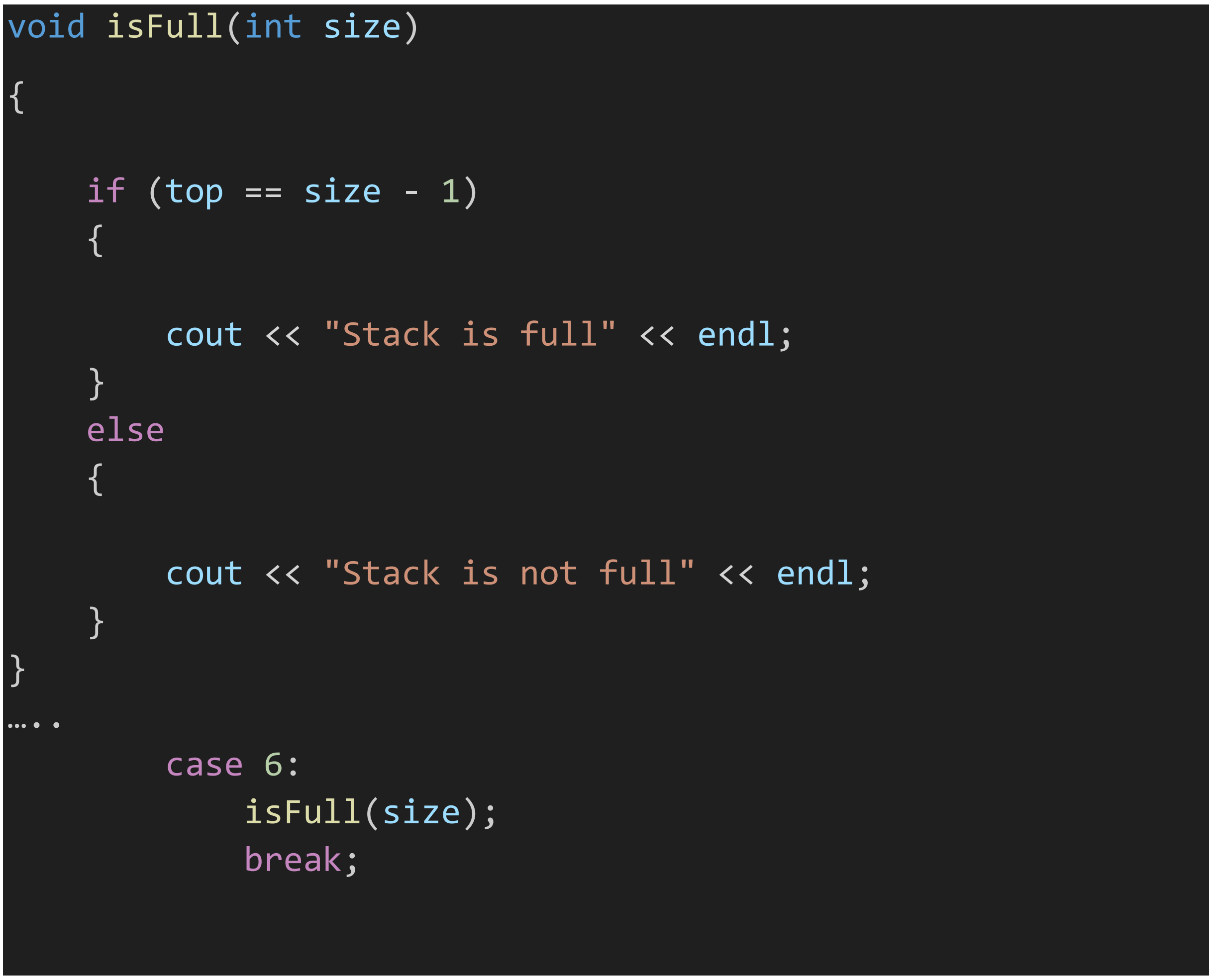
Stack Mechanism Discussion with Time Complexity

2. *IsFull*

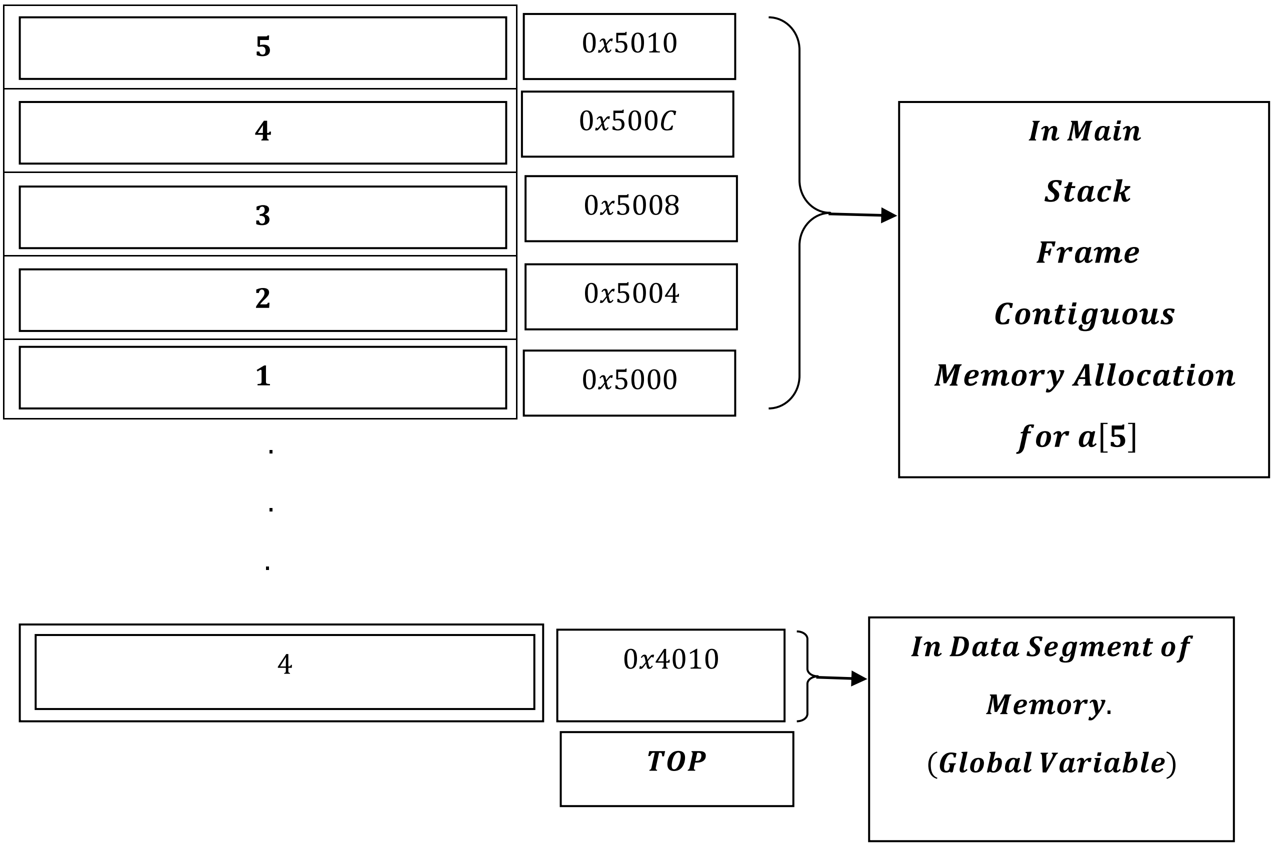


```
void isFull(int size)
{

    if (top == size - 1)
    {

        cout << "Stack is full" << endl;
    }
    else
    {

        cout << "Stack is not full" << endl;
    }
}
…..
        case 6:
            isFull(size);
            break;
```

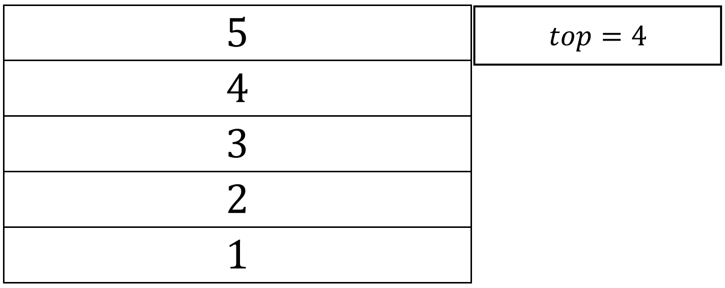
As top increments with $\boldsymbol{0, 1, 2, 3, \ldots. n - 1}$ ***, where n is size.***

Therefore when $\boldsymbol{top = size - 1}$ ***, then it outputs :***

``Stack is Full`` , else if false , then ``Stack is not Full``.


5
0x5010
4
0x500C
3
0x5008
2
0x5004
1
0x5000
In Main Stack Frame Contiguous Memory Allocation for a[5]
4
0x4010
TOP
In Data Segment of Memory. (Global Variable)

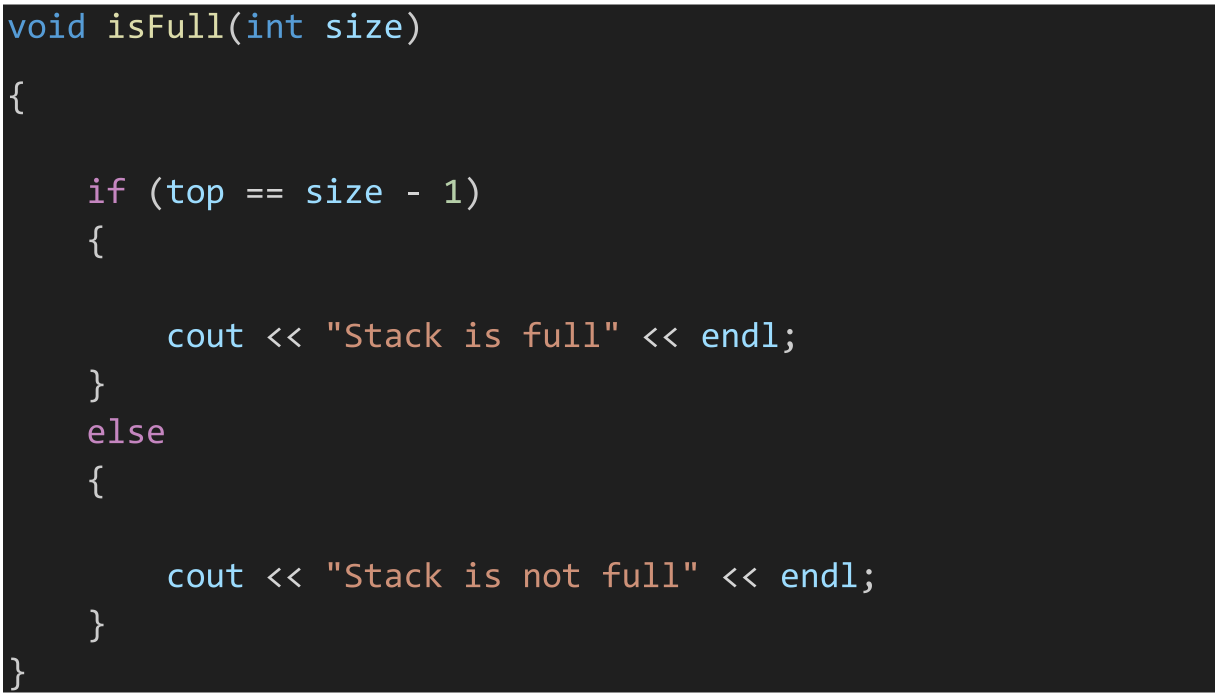
This is Physical Demonstration



Stack is Full

This is Logical Demonstration

Time Complexity



```
void isFull(int size)
{
    if (top == size - 1)
    {
        cout << "Stack is full" << endl;
    }
    else
    {
        cout << "Stack is not full" << endl;
    }
}
```

→ ***Function overhead or stack frame creation when isFull() is called takes constant time `c` takes*** $\boldsymbol{O}(1)$.

→ ***if*** $(\boldsymbol{top} = \boldsymbol{size} - 1)$ ***is true*** [***Takes constant time*** $\boldsymbol{O}(1)$]***then***:

Outputs: ``***Stack is Full***`` → ***also takes***: $\boldsymbol{O}(1)$.

If $(\boldsymbol{top} \neq \boldsymbol{size} - 1)$ ***then***:

Outputs: ``***Stack is not Full***`` → ***also takes***: $\boldsymbol{O}(1)$.

Total Time Complexity : $\boldsymbol{O}(1) + (\boldsymbol{O}(1) + (\boldsymbol{O}(1)) = \boldsymbol{O}(1)$.