**Name:** Safyan Anwar

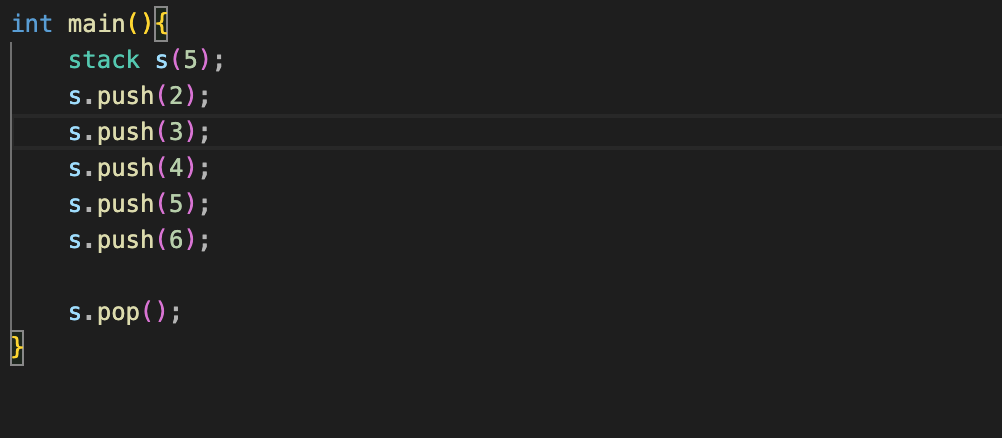
**Roll no:** 018

**Section:** 3A-BSSE

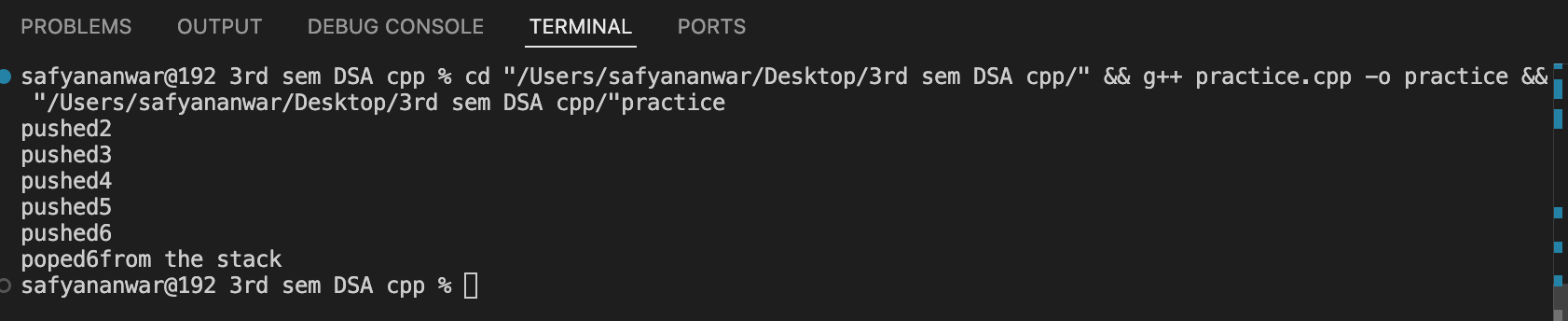
**LAB TASK:**

**Code: ‘**implement by array**’:-**

****

****

**Output:**

****

**Explanation:**

### **Class Members:**

* + **int \*arr:** A pointer to an integer array that will hold the stack elements.
  + i**nt size:** An integer that stores the maximum size of the stack.
  + **int top:** An integer that keeps track of the index of the top element in the stack. It starts at -1, indicating that the stack is empty.
* Public Members:
  + **Constructor:** Stack initializes the stack with a specified size s. It allocates memory for the array arr and sets top to -1.
  + **push(int value):** This method adds a new element to the top of the stack.
    - It first checks if the stack is full by comparing top with size - 1. If the stack is full, it prints stack overflow and returns.
    - If there is space, it increments top and assigns the value to then prints a message indicating the value has been pushed onto the stack.
  + **pop():** This method removes the top element from the stack.
    - It checks if the stack is empty by checking if top is -1. If it is empty, it prints stack underflow.
    - If there are elements in the stack, it prints the value being popped then decrements top to remove the element from the stack.

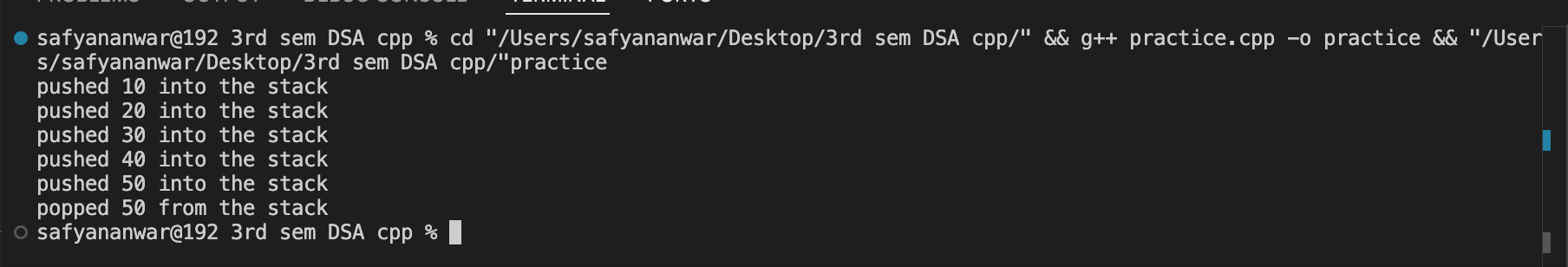
### **Main Function:**

* It then pushes five integers (2, 3, 4, 5, 6) onto the stack. The last push (6) will trigger a stack overflow message because the stack can only hold 5 elements.
* Finally, it calls the pop() method to remove the top element from the stack.

**Code: ‘**implemented code by linked list**’:-**

****

**Output:**

****

**Explanation:**

#### **Node Class:**

* **Members:**
  + **int data:** This variable holds the value of the node.
  + **Node\* next:** This is a pointer to the next node in the linked list.
* **Constructor:** The constructor initializes the data member with the provided value and sets next to NULL.

#### **Stack Class:**

* **Members:**
  + **Node top:** A pointer to the top node of the stack. It starts as NULL, indicating that the stack is empty.
  + **int size:** An integer that keeps track of the number of elements in the stack. It starts at 0.
* **Constructor:** The constructor initializes top to NULL and size to 0.

### **Stack Operations**

* **push(int d):**
  + A new Node is created with the value d.
  + It checks if the memory allocation for the new node was successful. If not, it prints stack overflow.
  + If the allocation is successful, the new node's next pointer is set to the current top, and then top is updated to point to the new node. The size is incremented, and a message is printed indicating the value that was pushed onto the stack.
* **pop():**
  + It checks if the stack is empty by checking if top is NULL. If it is empty, it prints stack underflow.
  + If there are elements in the stack, it stores the current top in a temporary variable prints the value being popped, updates top to point to the next node in the stack, and then deletes the temporary node to free the memory.

### **Main Function:**

* It pushes five integers (10, 20, 30, 40, and 50) onto the stack.
* Finally, it calls the pop() method to remove the top element from the stack.