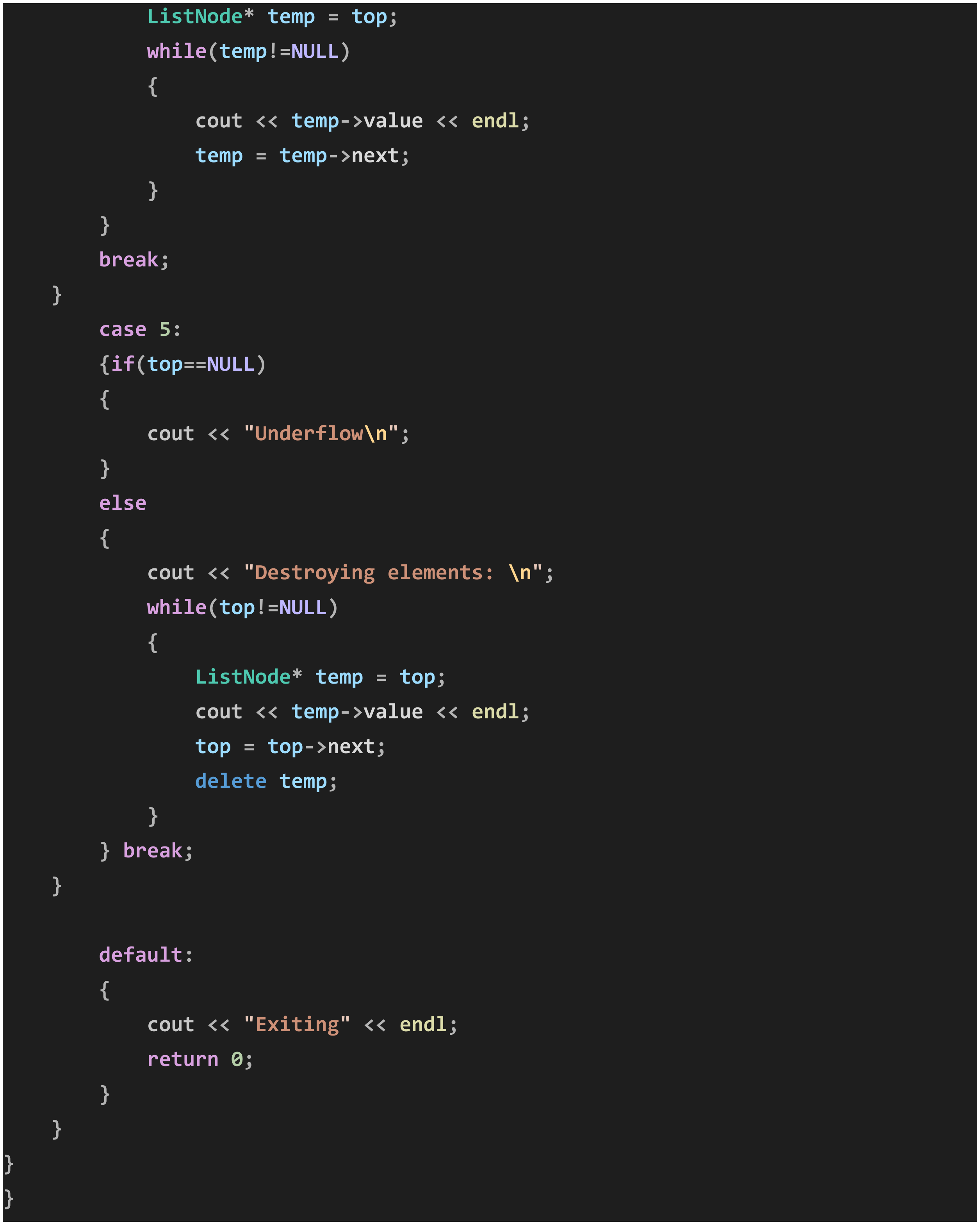
Code:

```
#include <bits/stdc++.h>
using namespace std;

struct ListNode
{
    int value;
    ListNode* next;
};

int main()
{
    ListNode* top = nullptr;
    cout << "Enter size of stack:" << endl;
    int n;
    cin >> n;
    while(true)
    {
    cout << "Options: 1. Push\n2. Pop\n3. Peek\n4. Display\n5. Destroy\nPress any other key to exit.\n";
    int choice;
    cin >> choice;
    switch(choice)
    {
        case 1:

            {ListNode* newnode = new ListNode();
            if(newnode == NULL)
            {
                cout << "Overload\n";
            }
            else
            {
                cout << "Push element:\n";
                int el;
                cin >> el;
                newnode->value = el;
                newnode->next = top; //points to the nullptr initially, and the topmost element lateron
                top = newnode;
            }
            break;

        }
        case 2:
        if(top==NULL)
        {
            cout << "Underflow\n";
        }
        else
        {
            cout << "Popped element: ";
            ListNode* temp = top;
            cout << temp->value << endl;
            top = top->next;
            delete temp;
        }break;


        case 3:
        {if(top==NULL)
        {
            cout << "Underflow\n";
        }
        else
        {
            cout << "Peeked at element: ";
            cout << top->value << endl;
        }
        break;}
        case 4:
        {if(top==NULL)
        {
            cout << "Underflow\n";
        }
        else
        {
            cout << "Displaying elements: \n";
```



```
            ListNode* temp = top;
            while(temp!=NULL)
            {
                cout << temp->value << endl;
                temp = temp->next;
            }
        }
        break;
    }
        case 5:
        {if(top==NULL)
        {
            cout << "Underflow\n";
        }
        else
        {
            cout << "Destroying elements: \n";
            while(top!=NULL)
            {
                ListNode* temp = top;
                cout << temp->value << endl;
                top = top->next;
                delete temp;
            }
        } break;
    }

        default:
        {
            cout << "Exiting" << endl;
            return 0;
        }
    }
}
}
```

Output:

```
PS C:\Users\syeda\OneDrive\Desktop\personal> cd
"c:\Users\syeda\OneDrive\Desktop\personal\" ; if ($?) { g++ ds.cpp -o ds } ; if ($?) { .\ds }
```

```
Enter size of stack:
5
Options: 1. Push
2. Pop
3. Peek
4. Display
5. Destroy
Press any other key to exit.
1
Push element:
5
Options: 1. Push
2. Pop
3. Peek
4. Display
5. Destroy
Press any other key to exit.
1
Push element:
6
Options: 1. Push
2. Pop
3. Peek
4. Display
5. Destroy
Press any other key to exit.
4
Displaying elements:
6
5
Options: 1. Push
2. Pop
3. Peek
4. Display
5. Destroy
Press any other key to exit.
3
Peeked at element: 6
Options: 1. Push
2. Pop
3. Peek
4. Display
5. Destroy
Press any other key to exit.
```

```
5
Destroying elements:
6
5
Options: 1. Push
2. Pop
3. Peek
4. Display
5. Destroy
Press any other key to exit.
4
Underflow
Options: 1. Push
2. Pop
3. Peek
4. Display
5. Destroy
Press any other key to exit.
L
Exiting
```