**Assignment 02**

**Name Muneeb Ahmad**

**Roll no 22011556-039**

**Section B**

**Submitted to Mr Azib Mehmood**

**Course title DSA**

**Course code IT-209**

**Department IT**

**Campus Hafiz Hayat Campus**



***OPERATIONS OF STACK***

#include<iostream>

#include<string>

using namespace std;

class Stack {

private:

int top;

int arr[5];

public:

// Constructor

Stack() {

top = -1; // First time stack is empty

for (int i = 0; i < 5; i++) {

arr[i] = 0;

}

}

**// (1) isEmpty function**

bool isEmpty() {

return top == -1;

}

**// (2) isFull function**

bool isFull() {

return top == 4; // Because array size is 5

}

**// (3) Push operation function**

void push(int val) {

if (isFull()) {

cout << "Stack overflow" << endl;

} else {

top++;

arr[top] = val;

}

}

**// (4) Pop operation function**

int pop() {

if (isEmpty()) {

cout << "Stack underflow" << endl;

return 0;

} else {

int popValue = arr[top];

arr[top] = 0;

top--;

return popValue;

}

}

**// (5) Count how many values in the stack**

int count() {

return (top + 1);

}

**// (6) Peek operation**

int peek(int pos) {

if (isEmpty() || pos < 0 || pos > top) {

cout << "Invalid position or stack underflow" << endl;

return 0;

} else {

return arr[pos];

}

}

**// (7) Changing the value inside the stack/array**

void change(int pos, int val) {

if (pos < 0 || pos > top) {

cout << "Invalid position" << endl;

} else {

arr[pos] = val;

cout << "Value changed at location " << pos << endl;

}

}

**// (8) Display stack elements**

void display() {

if (isEmpty()) {

cout << "The Stack is empty" << endl;

} else {

cout << "All the current values in the Stack are" << endl;

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

cout << arr[i] << endl;

}

}

}

**// (9) Search operation**

int search(int val) {

for (int i = 0; i <= top; i++) {

if (arr[i] == val) {

return i;

}

}

return -1; // Not found

}

**// (10) Update at any position**

void UAP(int pos, int val) { // UAP = update\_At\_Position

if (pos < 0 || pos > top) {

cout << "Invalid position" << endl;

} else {

arr[pos] = val;

cout << "Value updated at position " << pos << endl;

}

}

**// (11) Insert at any position**

void IAP(int pos, int val) { // IAP = insert\_At\_Position

if (isFull()) {

cout << "Stack overflow" << endl;

} else {

if (pos < 0 || pos > top + 1) {

cout << "Invalid position" << endl;

} else {

top++;

for (int i = top; i > pos; i--) {

arr[i] = arr[i - 1];

}

arr[pos] = val;

cout << "Value inserted at position " << pos << endl;

}

}

}

**// (12) Delete from beginning**

void dfb() { // dfb = delete\_From\_Beginning

if (isEmpty()) {

cout << "Stack underflow" << endl;

} else {

for (int i = 0; i < top; i++) {

arr[i] = arr[i + 1];

}

arr[top] = 0;

top--;

cout << "Deleted from beginning" << endl;

}

}

**// (13) Delete from end**

void dfe() { // dfe = delete\_From\_End

if (isEmpty()) {

cout << "Stack underflow" << endl;

}

else {

arr[top] = 0;

top--;

cout << "Deleted from end" << endl;

}

}

**// (14) Delete from any position**

void dfp(int pos) { // dfp = deleteFromPosition

if (isEmpty()) {

cout << "Stack underflow" << endl;

} else {

if (pos < 0 || pos > top) {

cout << "Invalid position" << endl;

} else {

for (int i = pos; i < top; i++) {

arr[i] = arr[i + 1];

}

arr[top] = 0;

top--;

cout << "Deleted from position " << pos << endl;

}

}

}

};

int main() {

Stack s1;

int option, position, value;

// Now i am creating menu

do {

cout << "What operation do you want to perform? Select Option number. Enter 0 to exit." << endl;

cout << "1. Push()" << endl;

cout << "2. Pop()" << endl;

cout << "3. isEmpty()" << endl;

cout << "4. isFull()" << endl;

cout << "5. peek()" << endl;

cout << "6. count()" << endl;

cout << "7. change()" << endl;

cout << "8. display()" << endl;

cout << "9. Search()" << endl;

cout << "10. Update at any position" << endl;

cout << "11. Insert at any position" << endl;

cout << "12. Delete from beginning" << endl;

cout << "13. Delete from end" << endl;

cout << "14. Delete from any position" << endl;

cout << "15. Clear Screen" << endl << endl;

cin >> option;

switch (option) {

case 0:

break;

case 1:

cout << "Please enter any item to push in the stack" << endl;

cin >> value;

s1.push(value);

break;

case 2:

cout << "Now calling Pop Function - Popped Value: " << s1.pop() << endl;

break;

case 3:

if (s1.isEmpty())

cout << "Stack is Empty" << endl;

else

cout << "Stack is not Empty" << endl;

break;

case 4:

if (s1.isFull())

cout << "Stack is Full" << endl;

else

cout << "Stack is not Full" << endl;

break;

case 5:

cout << "Please enter position of those item you want to peek: " << endl;

cin >> position;

cout << "Now calling Peek Function - Value at position " << position << " is " << s1.peek(position) << endl;

break;

case 6:

cout << "Now calling Count Function - Number of Items in the Stack are: " << s1.count() << endl;

break;

case 7:

cout << "Now calling Change Function - " << endl;

cout << "Please enter position of those items you want to change : ";

cin >> position;

cout << endl;

cout << "Please enter value of item you want to change : ";

cin >> value;

s1.change(position, value);

break;

case 8:

cout << "Now calling display Function - " << endl;

s1.display();

break;

case 9:

cout << "Please enter the value you want to search: ";

cin >> value;

position = s1.search(value);

if (position != -1)

cout << "the Value you enetred found at position " << position << endl;

else

cout << "the value you entered is not found" << endl;

break;

case 10:

cout << "Please enter a position to update: ";

cin >> position;

cout << "Please enter a new value: ";

cin >> value;

s1.UAP(position, value);

break;

case 11:

cout << "Please enter any position to insert: ";

cin >> position;

cout << "Please enter any value to insert: ";

cin >> value;

s1.IAP(position, value);

break;

case 12:

s1.dfb();

break;

case 13:

s1.dfe();

break;

case 14:

cout << "Please enter any position to delete: ";

cin >> position;

s1.dfp(position);

break;

case 15:

system("cls");

break;

default:

cout << "invalid! Please enter Proper Option number " << endl;

}

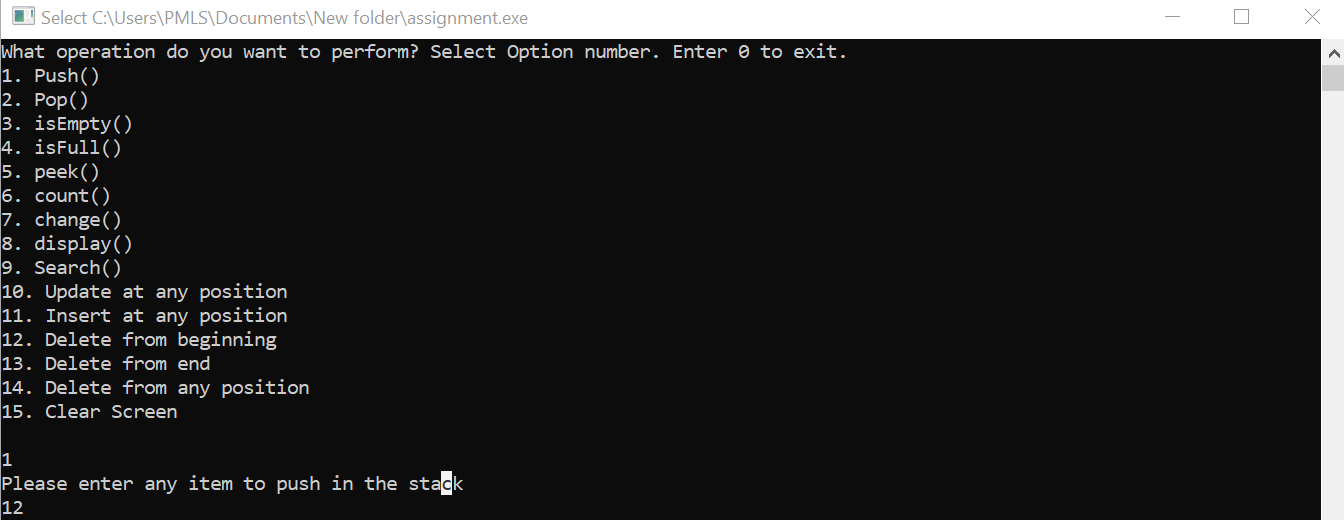
} while (option != 0);

return 0;

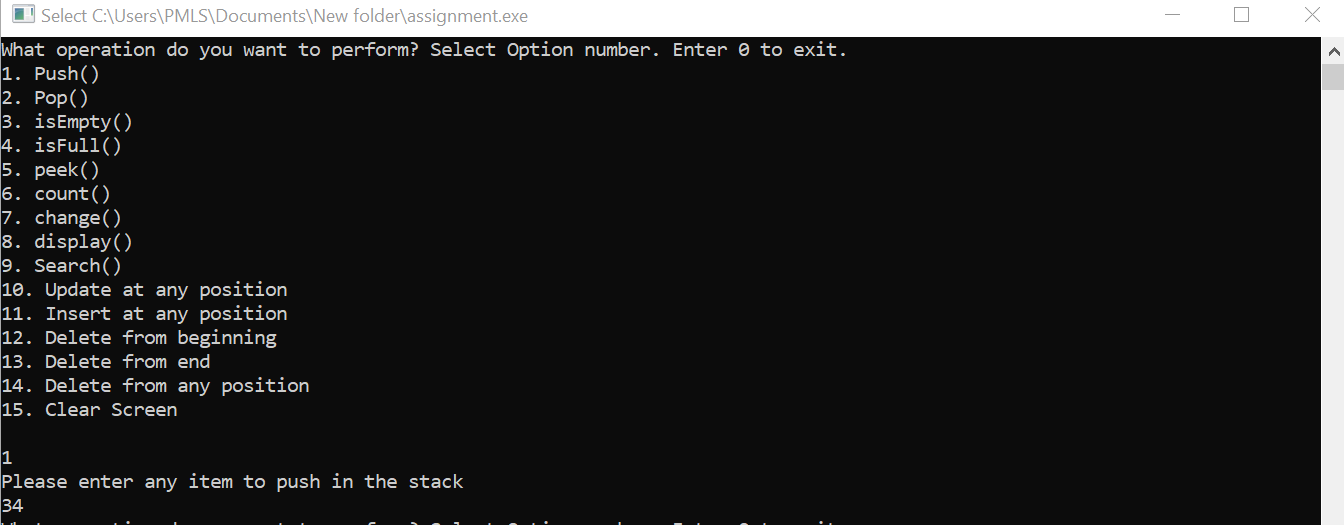
}

***OUTPUT***

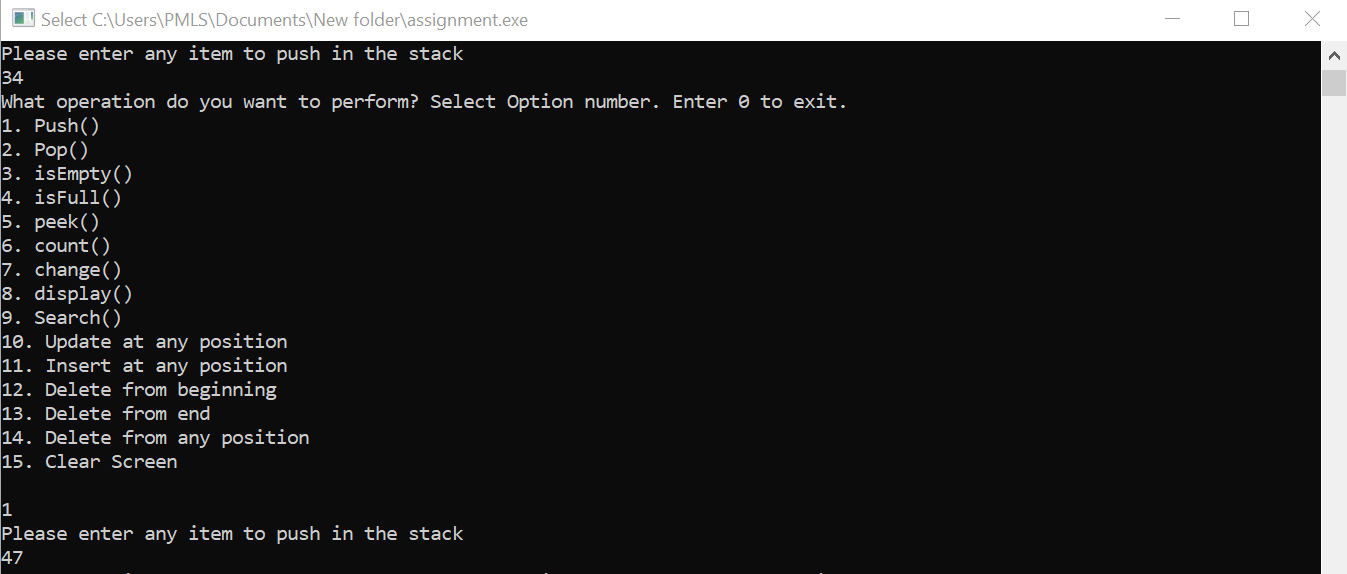
Selecting Option 1 to push first value:



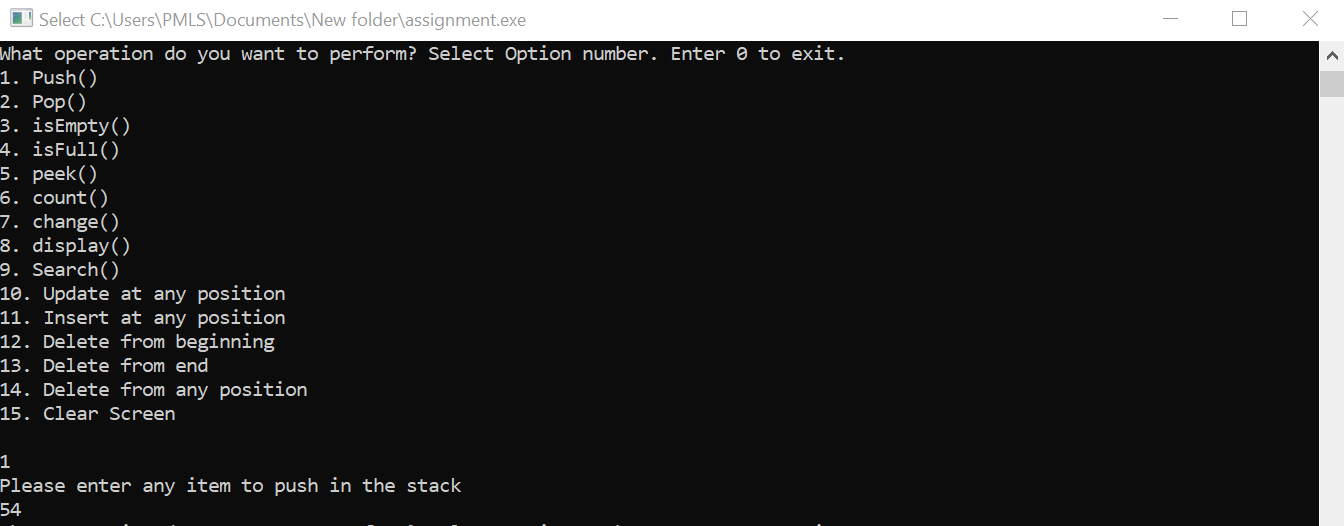
Selecting option 1 to to push second value :



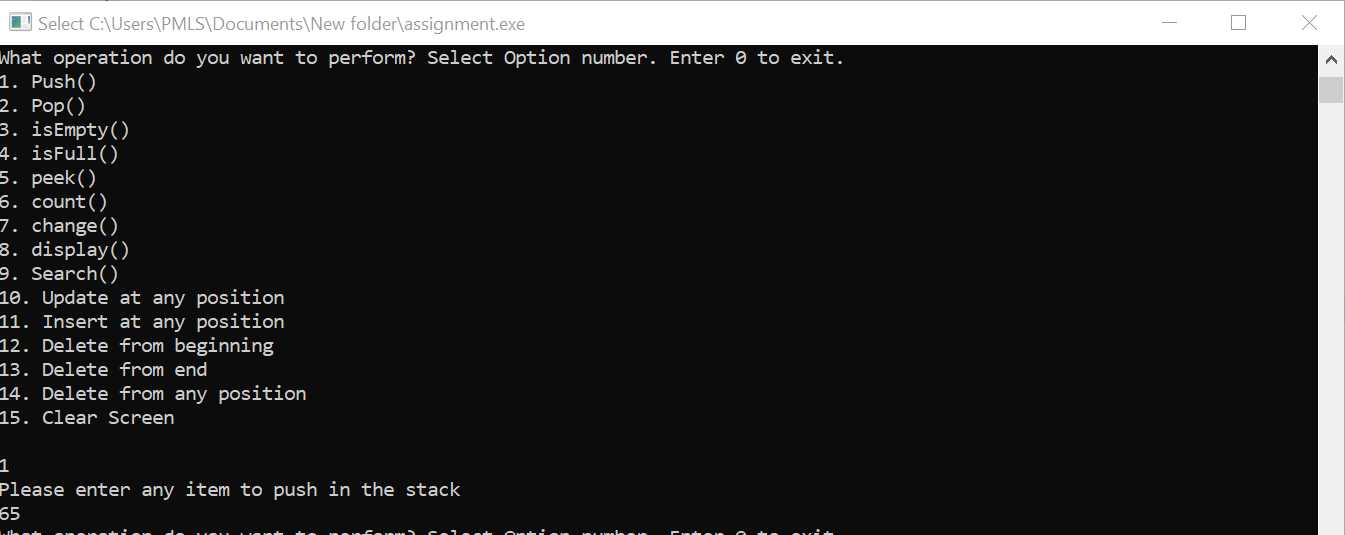
Selecting option 1 to push third value:



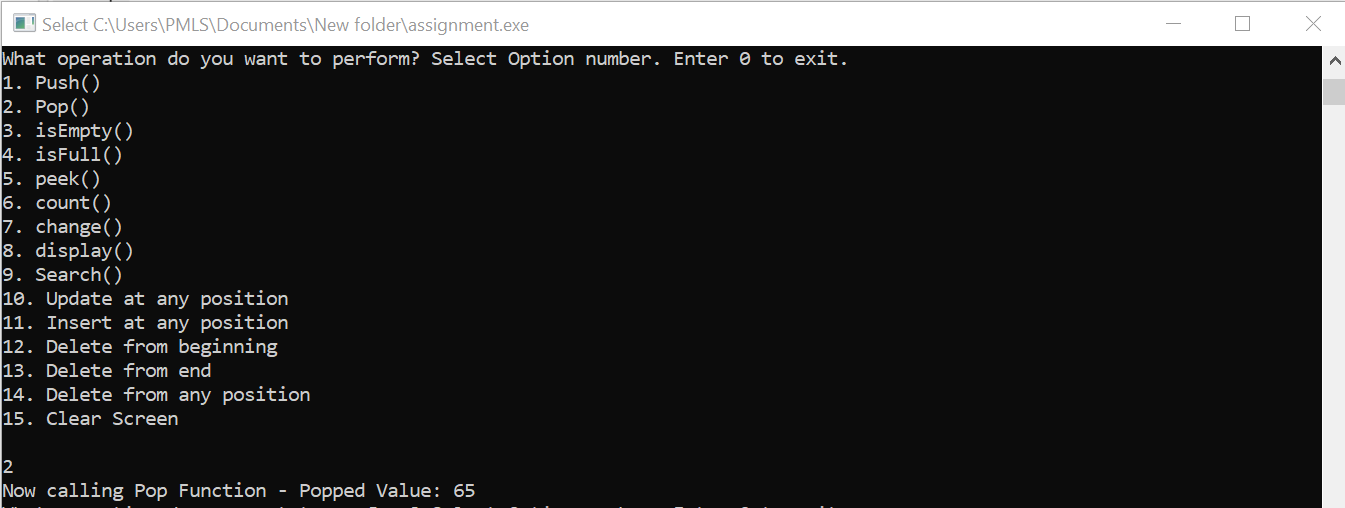
Selecting option 1 to push fourth value:



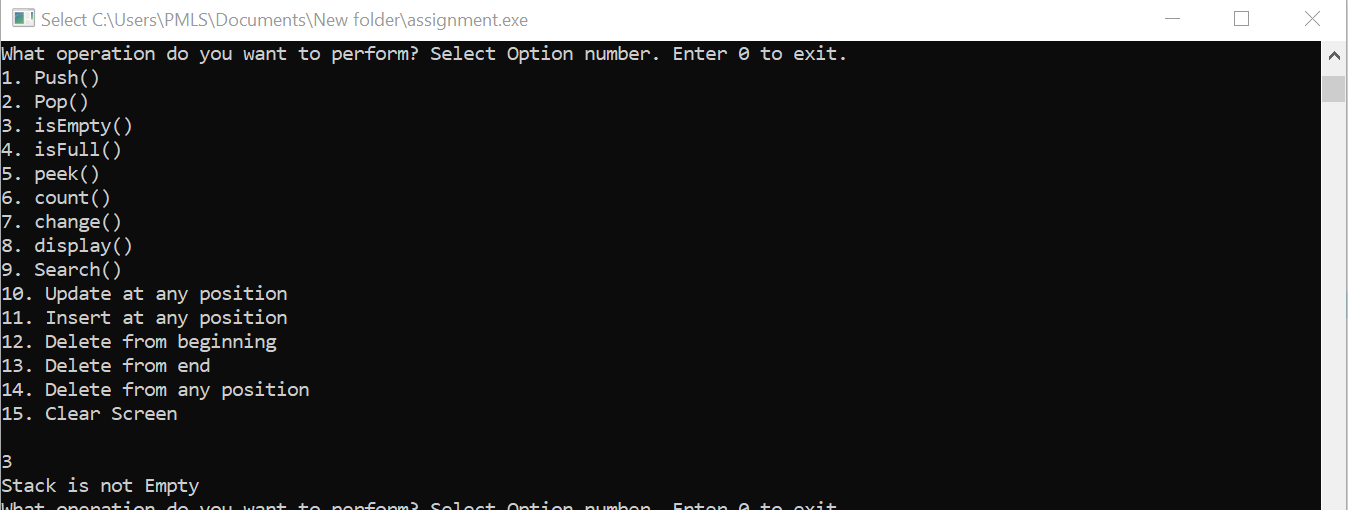
Selecting option 1 to push fifth value:



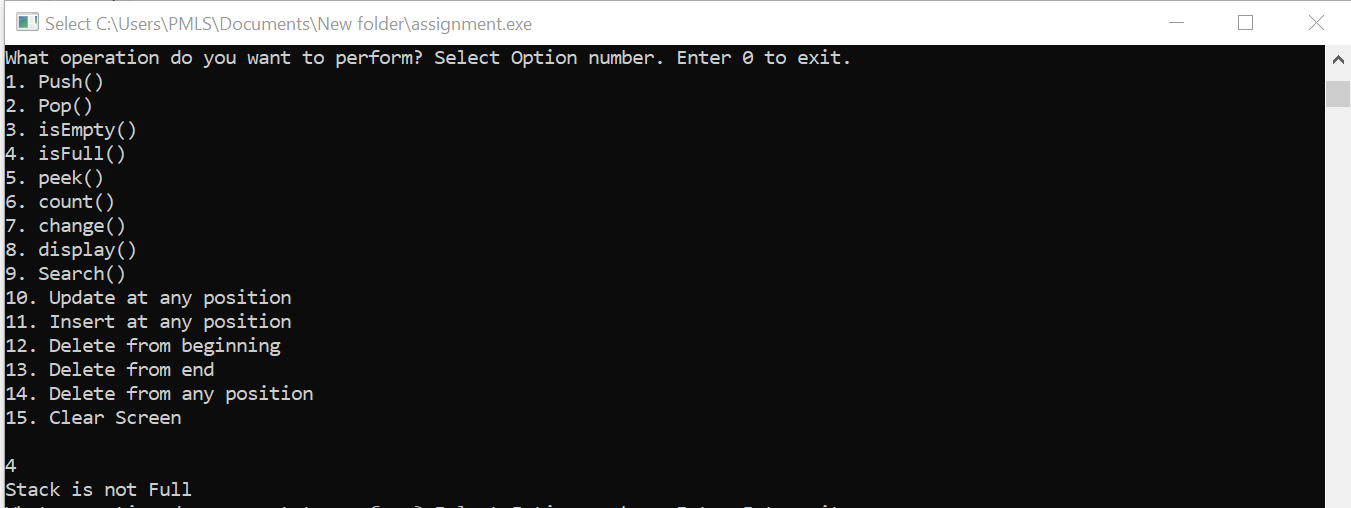
Selecting Option 2 to pop a value :



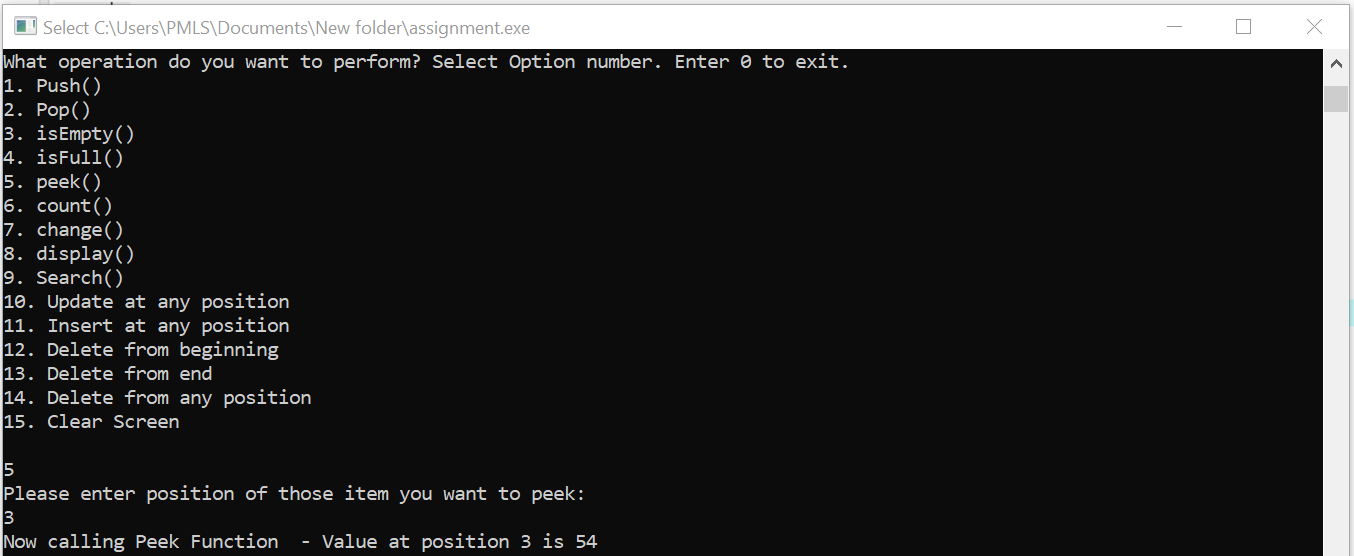
Selecting option 3 to check if stack is empty or not:



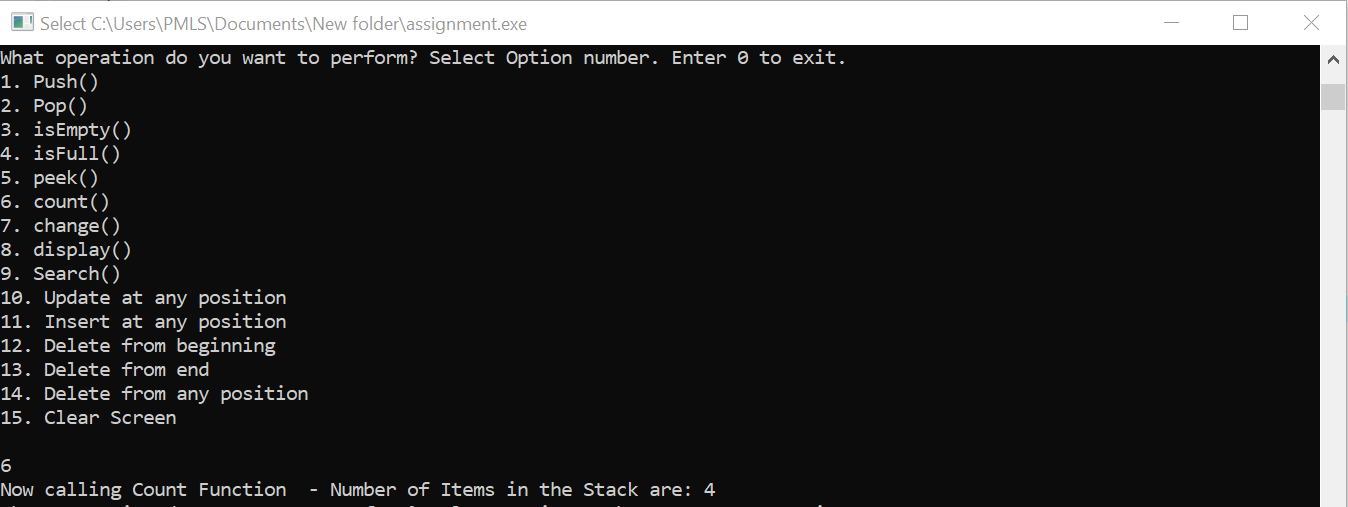
Selecting option 4 to check if stack is full or not :



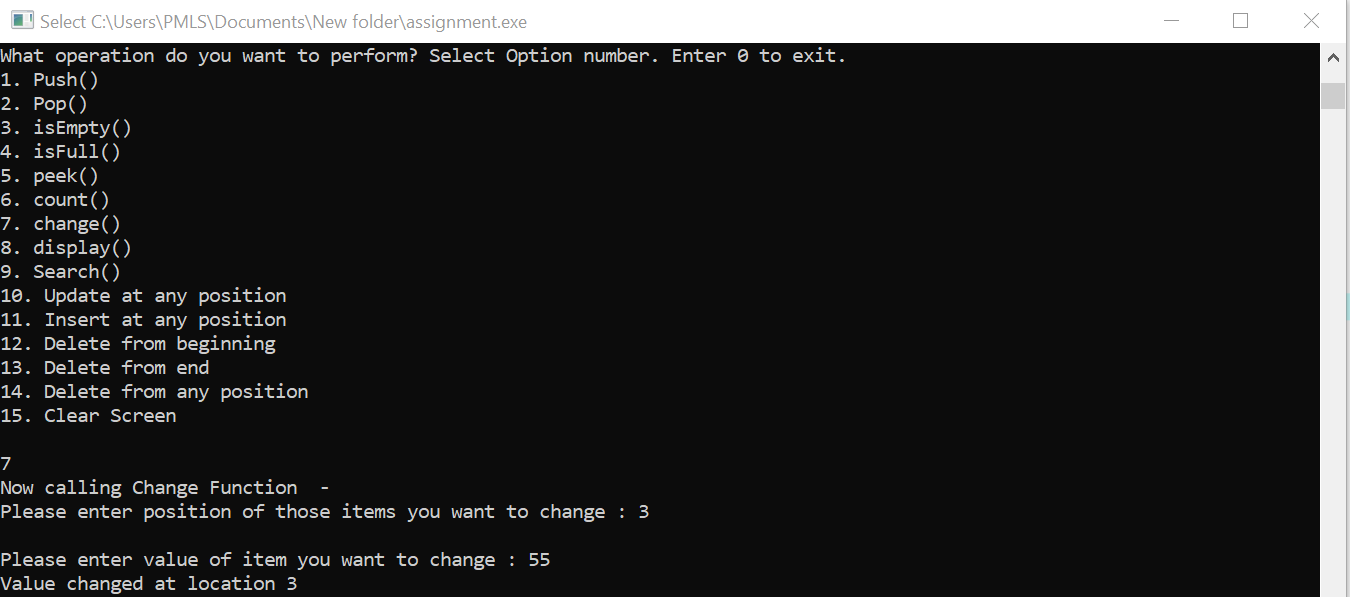
Selecting option 5 to allow inspection of element at a particular position:



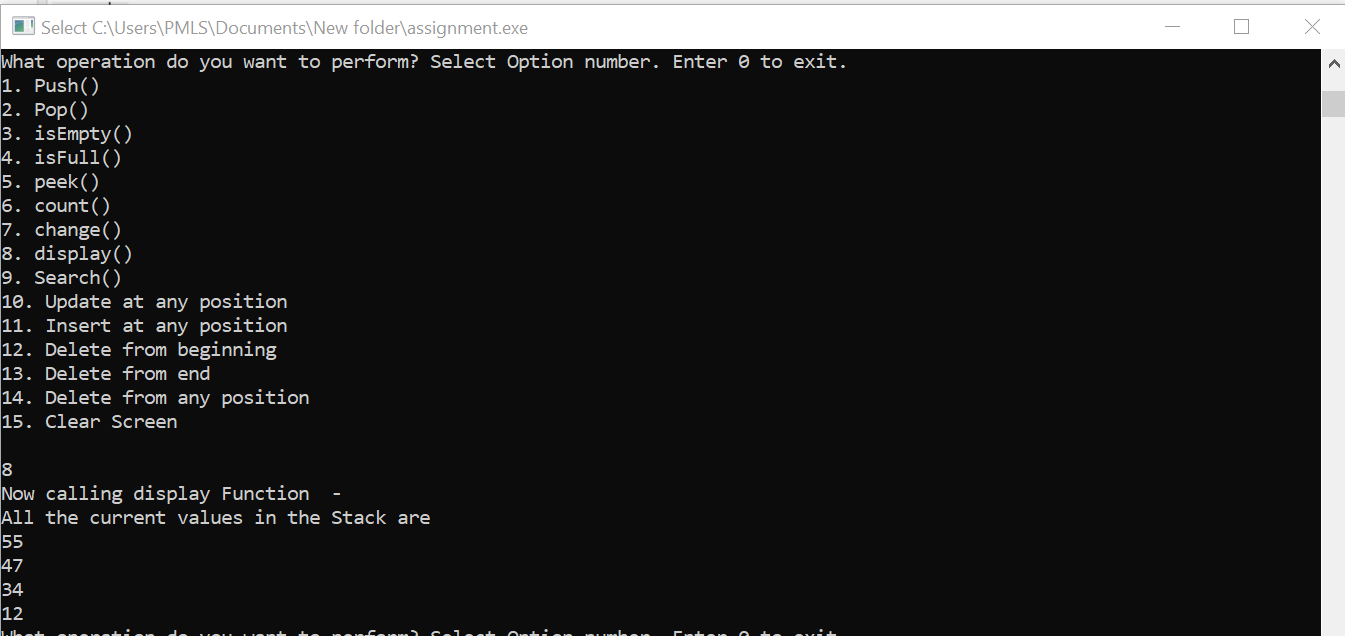
Selecting option 6 to count the total number of items in stack:



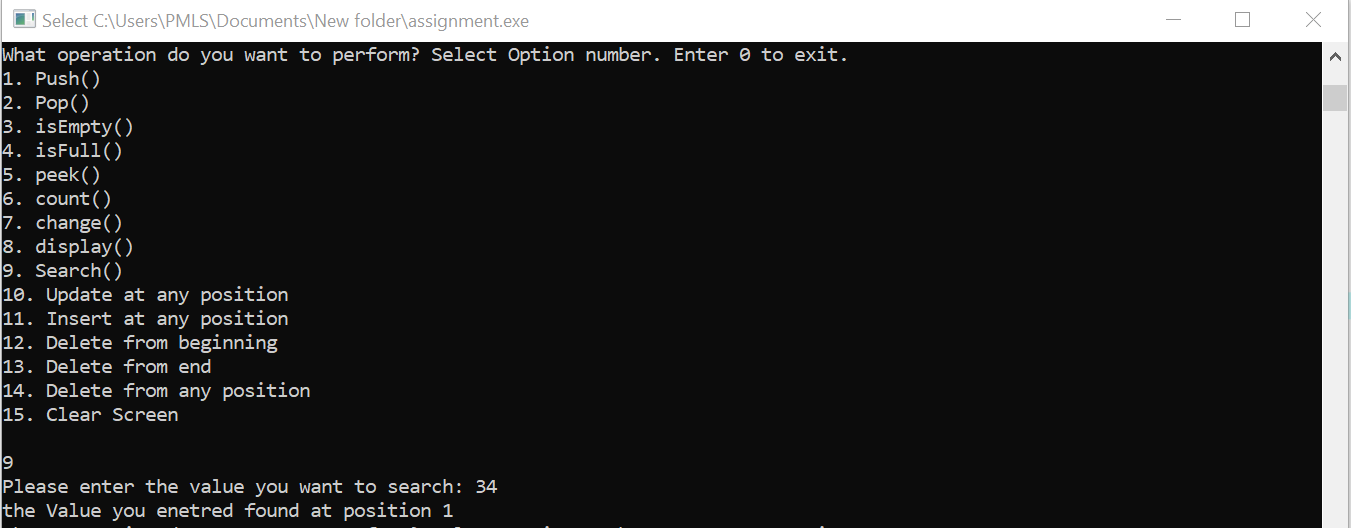
Selecting option 7 to change the item in the stack:



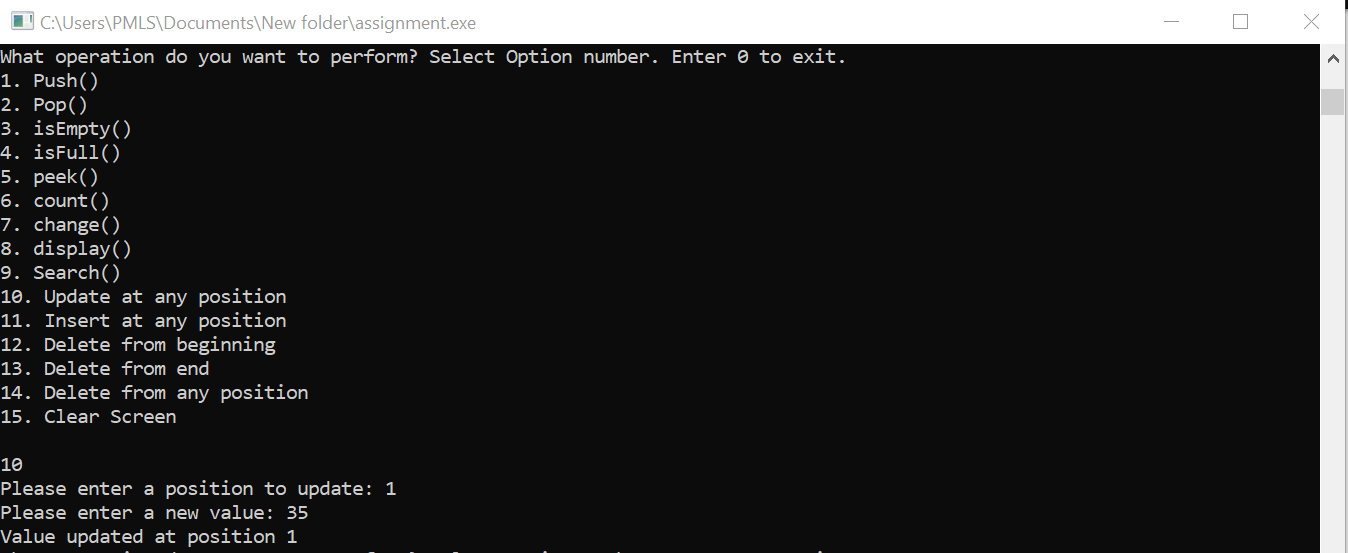
Selecting option 8 to display the items :



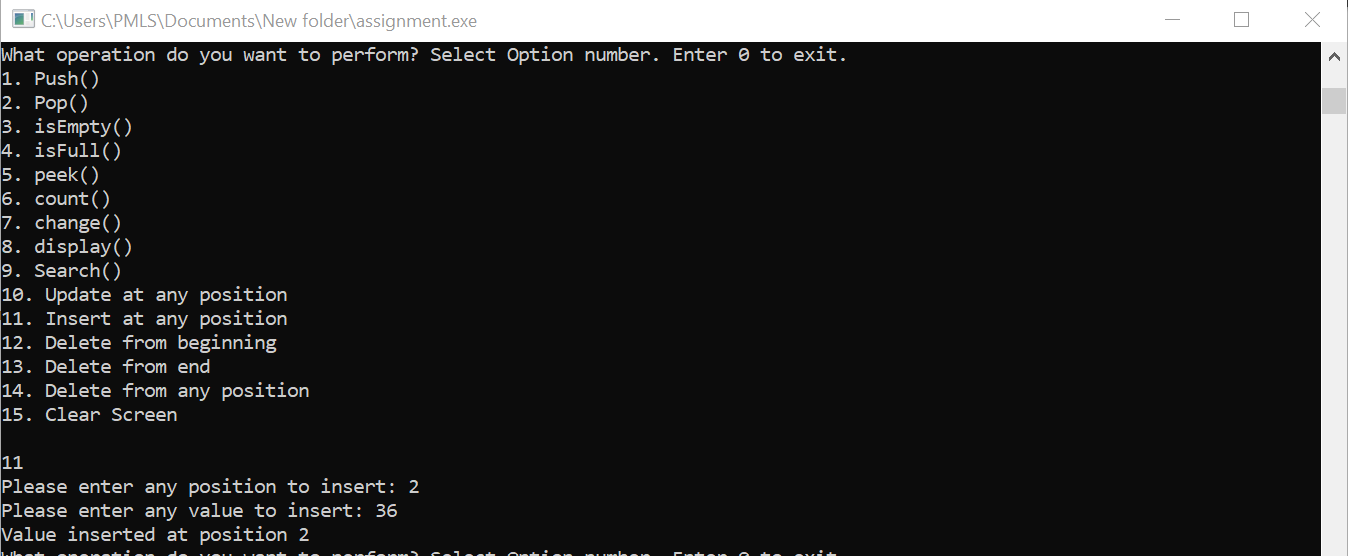
Selecting option 9 to search item from the stack:



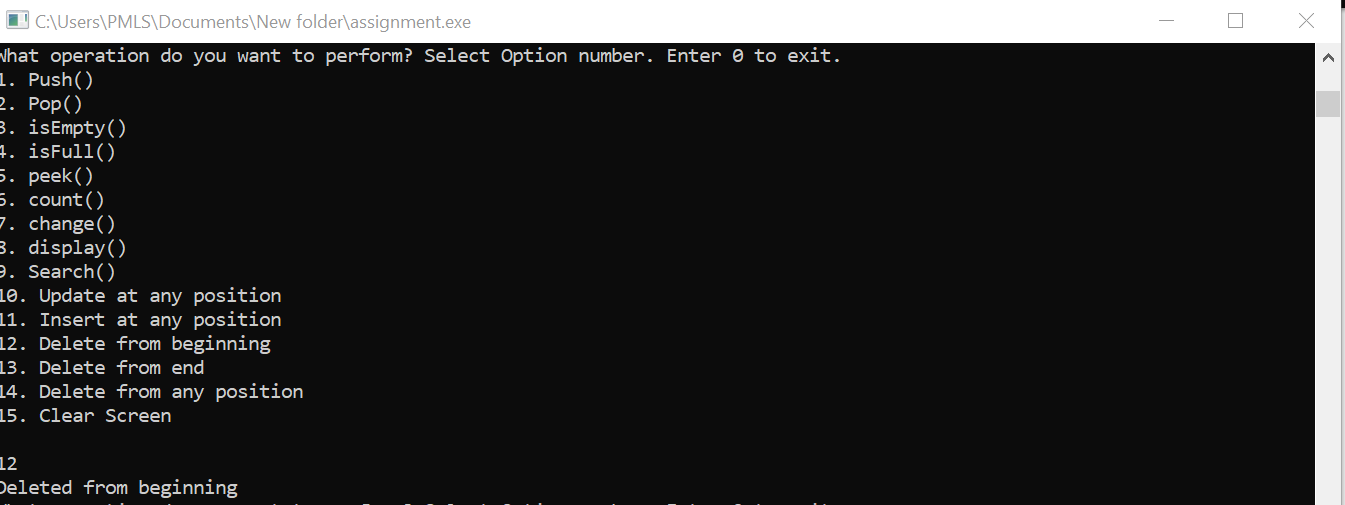
Selecting option 10 to update item at any position in the stack:



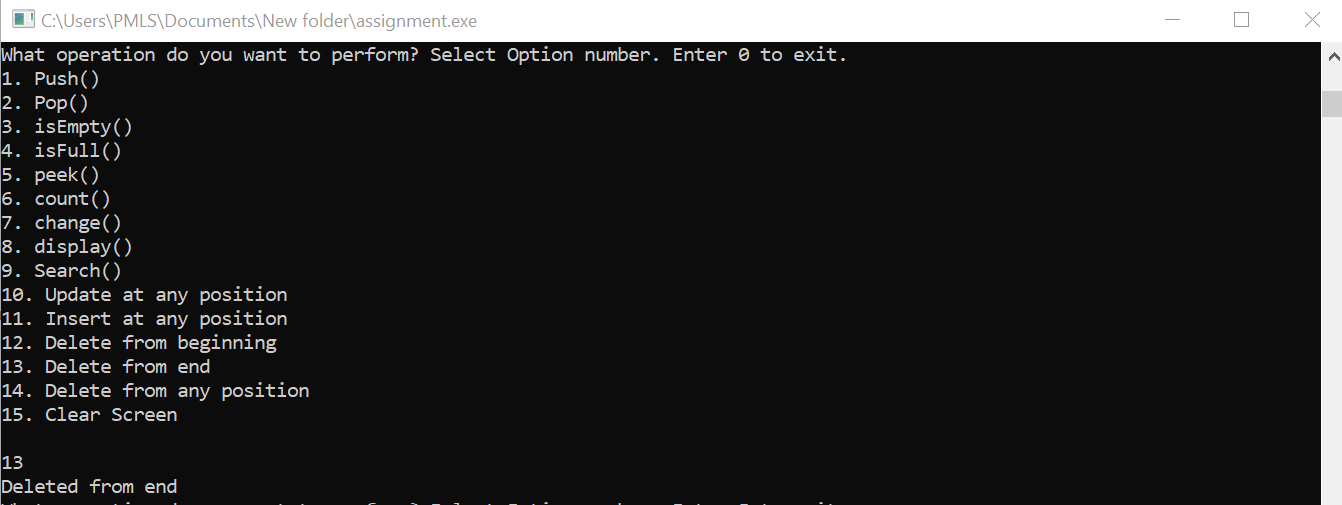
Selecting option 11 to insert item at any position in the stack:



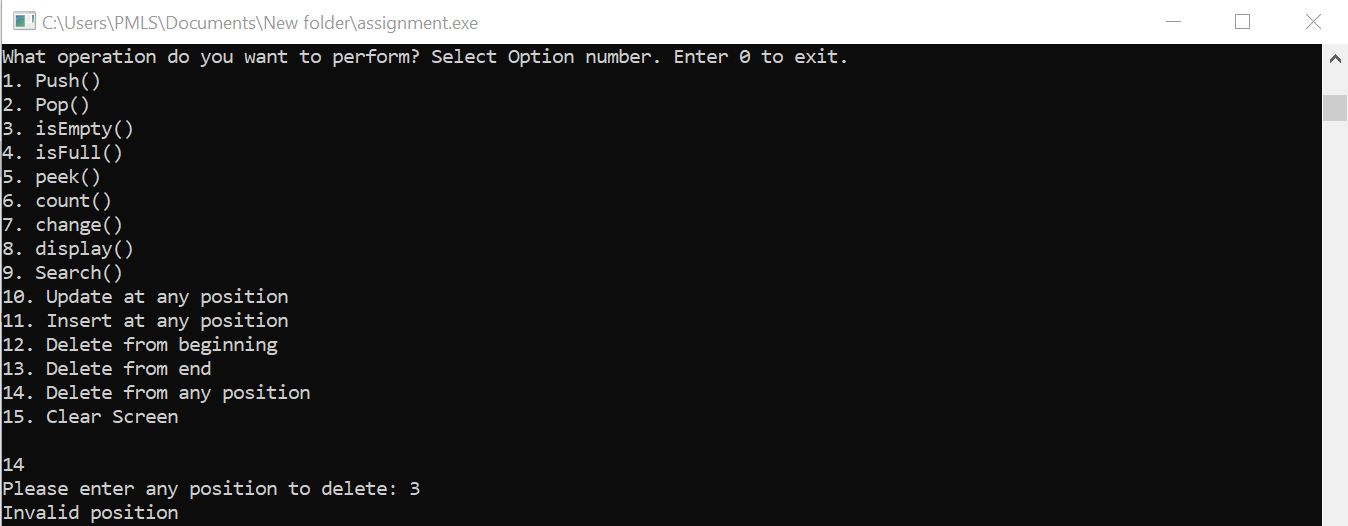
Selecting option 12 to delete item from the beginning:



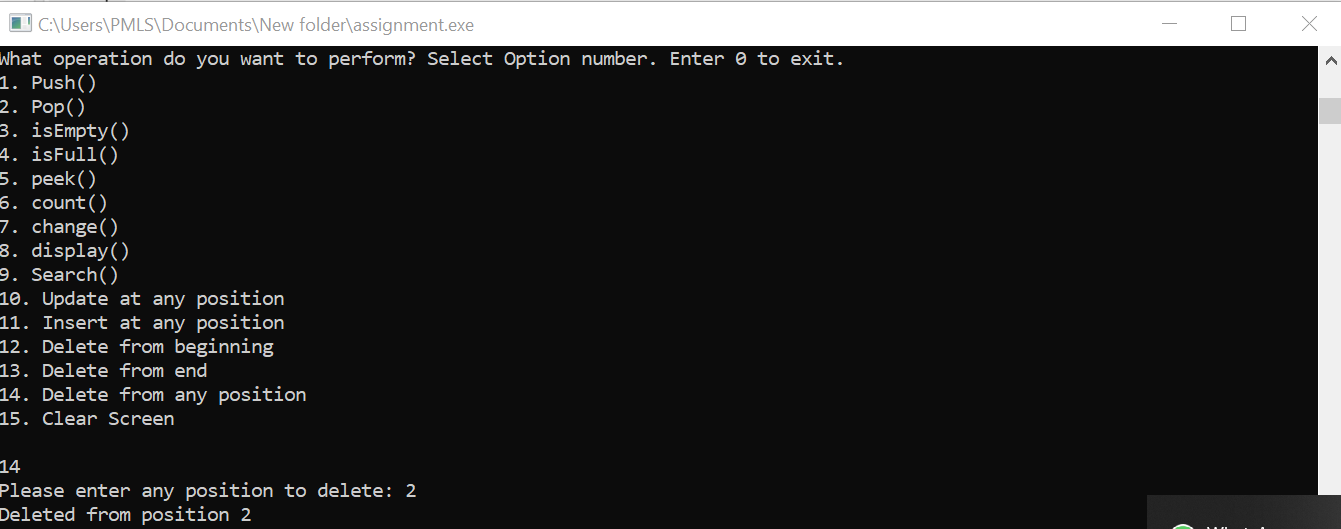
Selecting option 13 to delete item from the end:



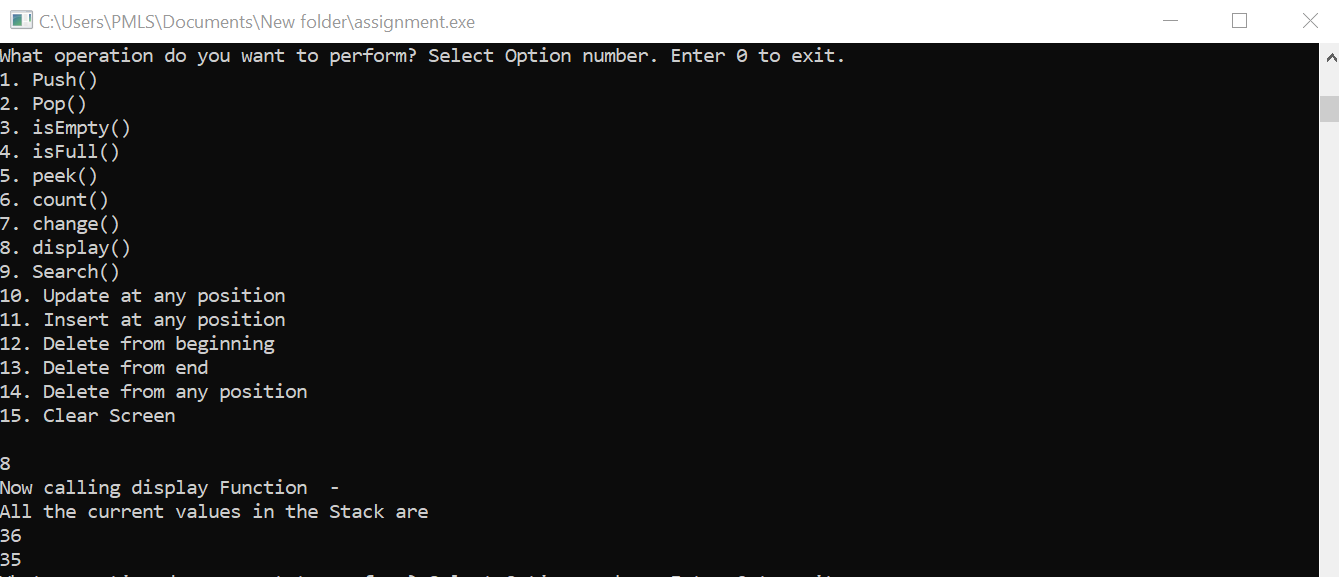
Selecting option 14 to delete item from any location :



Selecting option 14 to delete item from any location :



Selecting option 8 to display the items left in the stack after performing these operations :



Selecting option 15 to clear the screen:

