Data Structure and Algorithm
Lab Experiment 1
Stack Array Implementation

Name: Om Ashish Mishra

Registration No.: 16BCE0789

Slot: G2

The Pseudo Code:

- First we use a macro to get define a variable SIZE of 5 memory space.
- Then we declare the pre-processor directive in order to write the header files.
- Then we declare a structure stack and take the array(s[SIZE]) and top to point to the top of the stack.
- We declare an object ‘st’ for the stack.
- Then the function isfull() and isempty() are made to check for ‘OVERFLOW’ or ‘UNDERFLOW’ of the stack.
- Then the function push() and pop() are defined to add or remove the elements from the array respectively.
- Then we enter the main function and the MENU-DRIVEN part comes.
- The item is used to take input into the stack and ch is used to take user’s choices.
- Top is initialized to -1.
- We ask the user he/she wants to push or pop an element or wants to display or exit from the program.
- Then the desired operation takes place.
- At last we ask if the user wants to continue the program again. If yes it does else exit.

The Code:

```
#include<stdio.h>

#include<stdlib.h>

#define SIZE 5          /* The macro is declared to SIZE 5 */

struct stack           /* Structure is made with an array s[size] and top to point the topmost element in the stack. */

{

  char s[SIZE];

  int top;

} st;                  /* Object of the structure */


int isfull()           /* To check whether the stack is full or not. */

{

  if (st.top >= SIZE - 1)

    return 1;

  else

    return 0;

}


int isempty()          /* To check the stack is empty or not. */

{

  if (st.top == -1)

    return 1;

  else
```

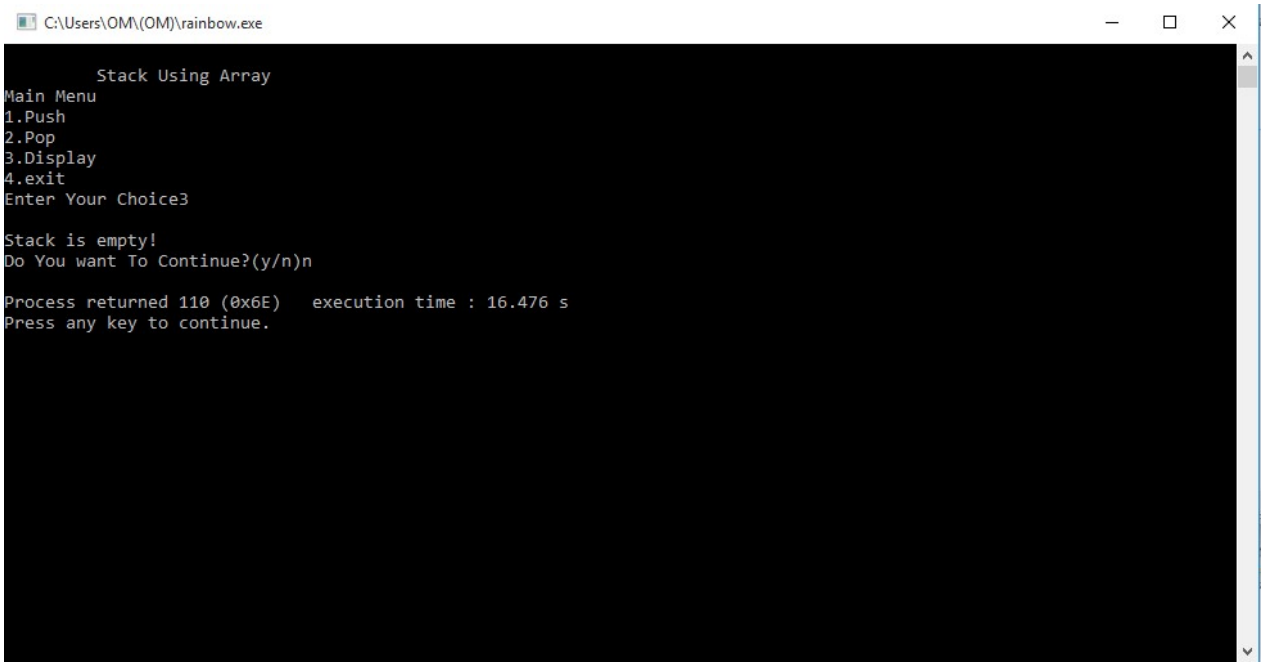
```
    return 0;
}

void push(char elem)    /* To push the elements into the stack. */
{
  st.top++;
  st.s[st.top] = elem;
}

int pop()             /* To delete the elements from the stack */
{
  char elem;
  elem = st.s[st.top];
  st.top--;
  return (elem);      /* This helps us to print the deleted item. */
}

void display()
{
  int i;
  if (isempty())
    printf("\nStack is empty!");
  else
  {
    for (i = st.top; i >= 0; i--)    /* Display of the elements. */
```

```
      printf("\n%d", st.s[i]);
  }
}

void main()
{
  char item;
  int ch;                 /* item is used take user's input into the stack and ch is used to take user's
choices.*/
  char u='y';             /* User's choice to enter into the program again. */
  st.top = -1;            /* Initializing the value of top. */

  printf("\n\t Stack Using Array");    /* MENU-DRIVEN program */
  do {
    printf("\nMain Menu");
    printf("\n1.Push \n2.Pop \n3.Display \n4.exit");
    printf("\nEnter Your Choice");
    scanf("%d", &ch);
    switch (ch)
    {
    case 1:                                          /*Insertion into stack.*/
      printf("\nEnter The item to be pushed");
      scanf("%s", &item);
      if (isfull())
        printf("\nStack is Full!");
      else
```

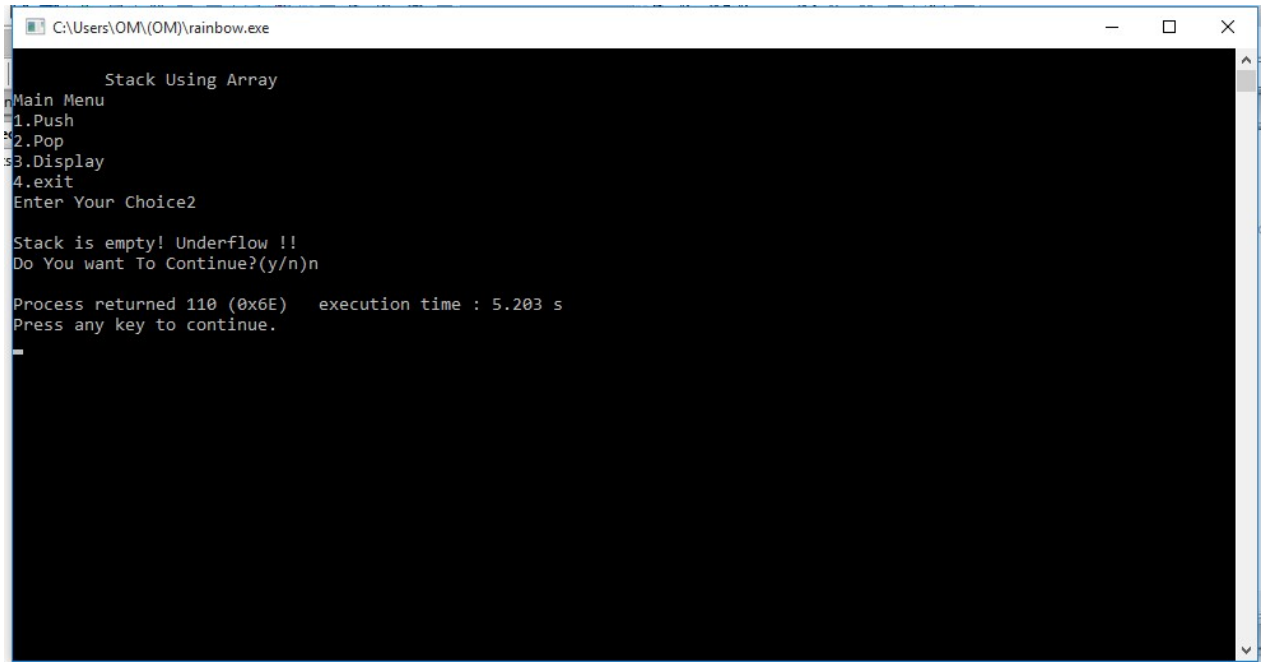
```
        push(item);
        break;
      case 2:                                   /*Deletion from stack*/
        if (isempty())
          printf("\nStack is empty! Underflow !!");
        else
        {
          item = pop();
          printf("\nThe popped element is %d", item);
        }
        break;
      case 3:                                   /*Displaying the elements*/
        display();
        break;
      case 4:
        printf("Wrong choice");
        exit(0);
      }
      printf("\nDo You want To Continue?(y/n)");
      scanf("%s",&u);
    } while (u == 'Y' || u == 'y');
}
```

The Outputs:

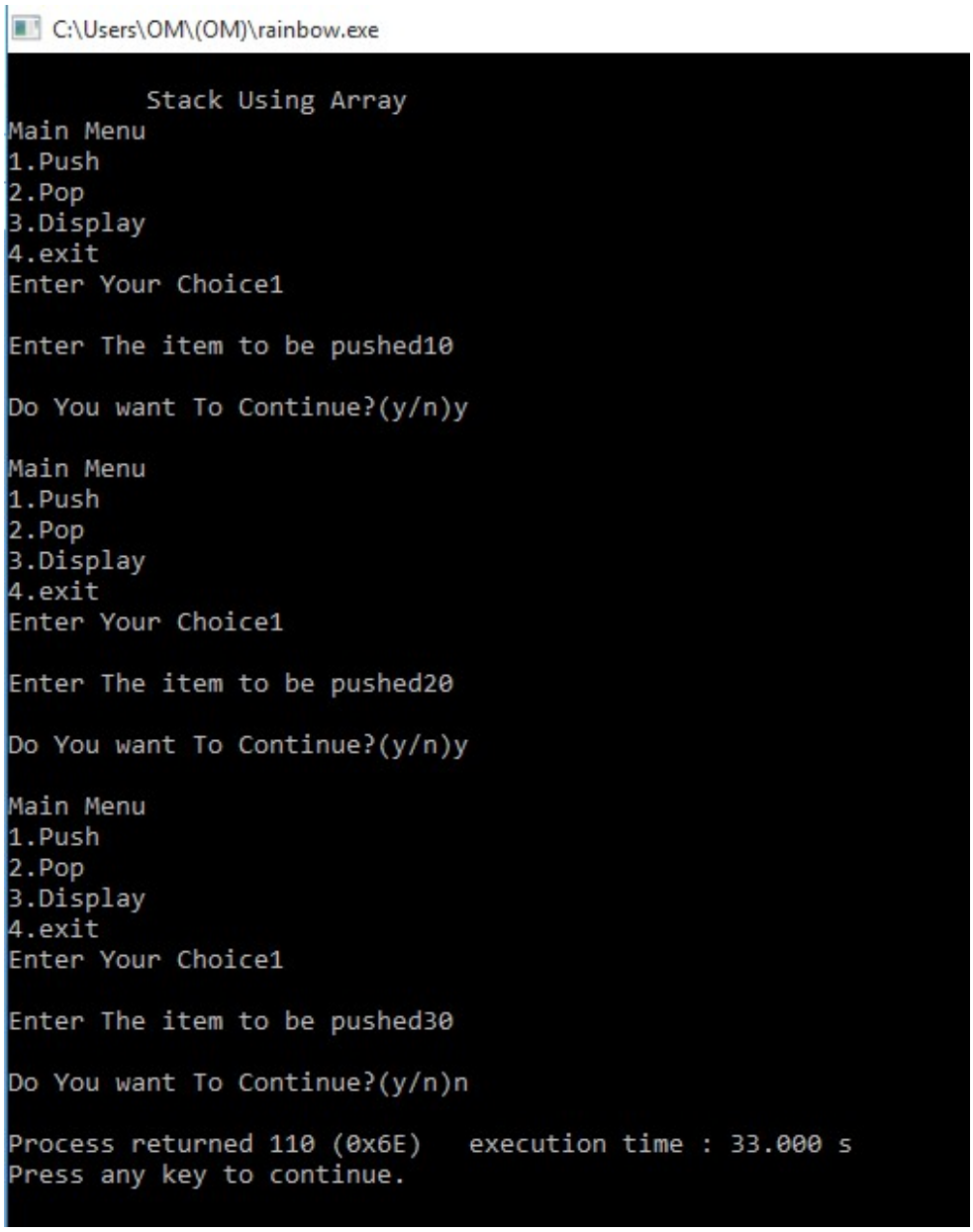
We check the display when the array is empty.



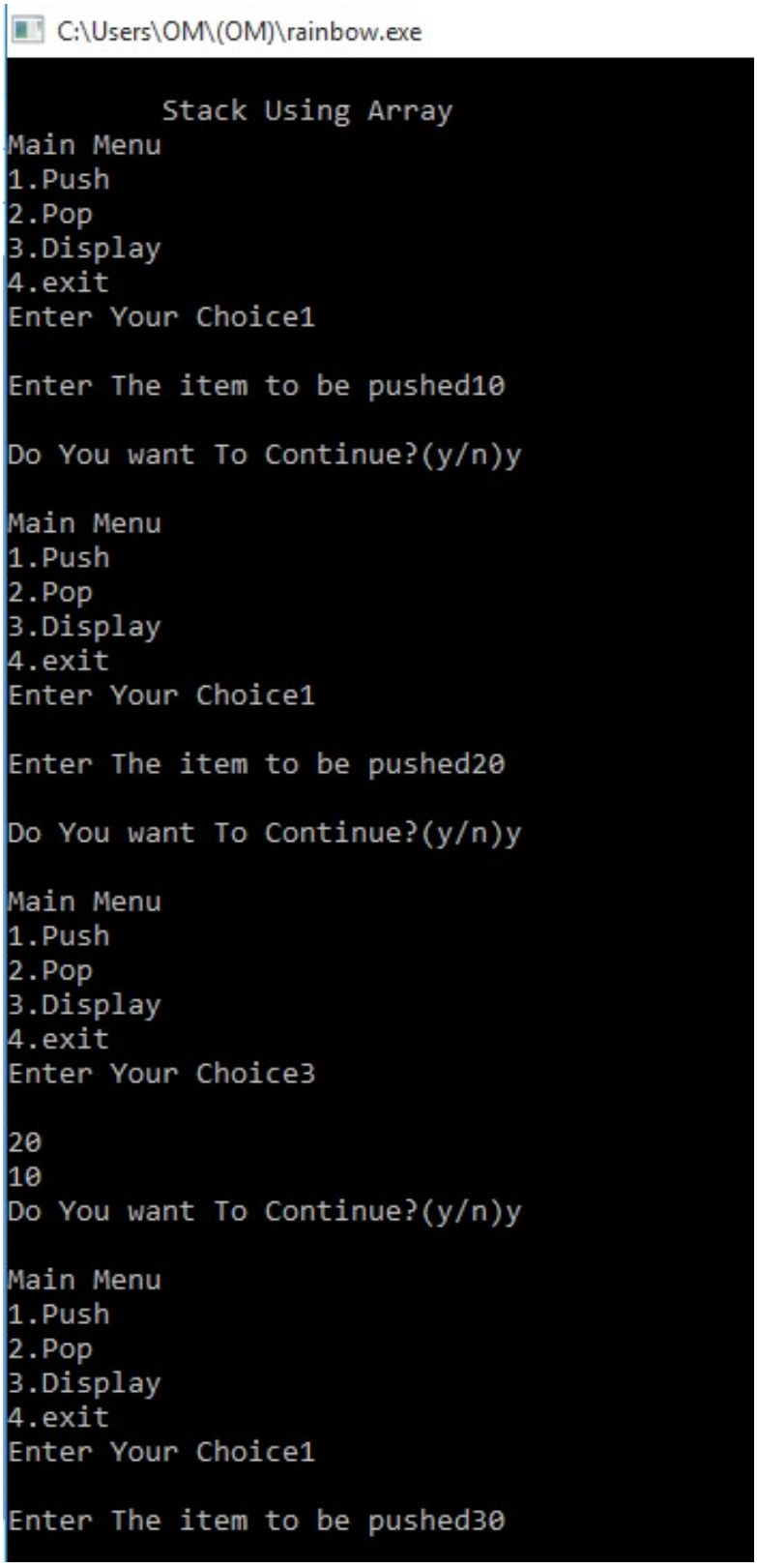
We check the pop when the array is empty.

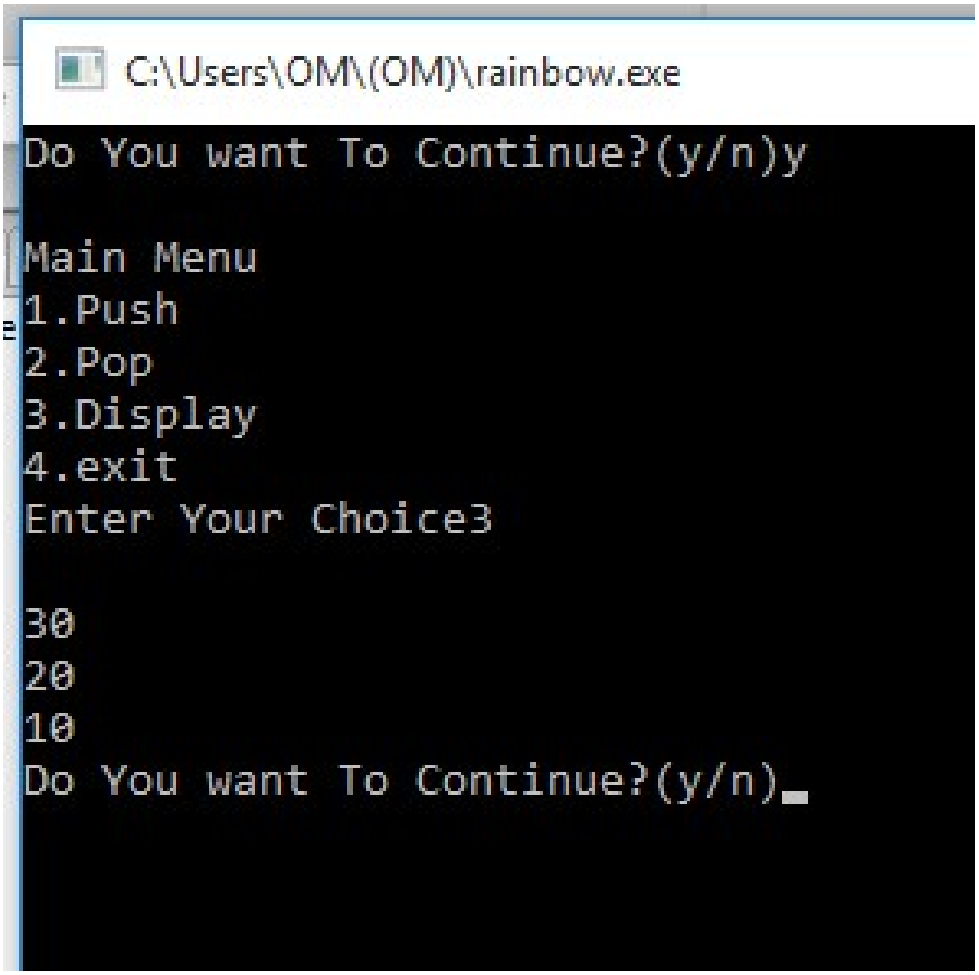


Now we put elements into the array.

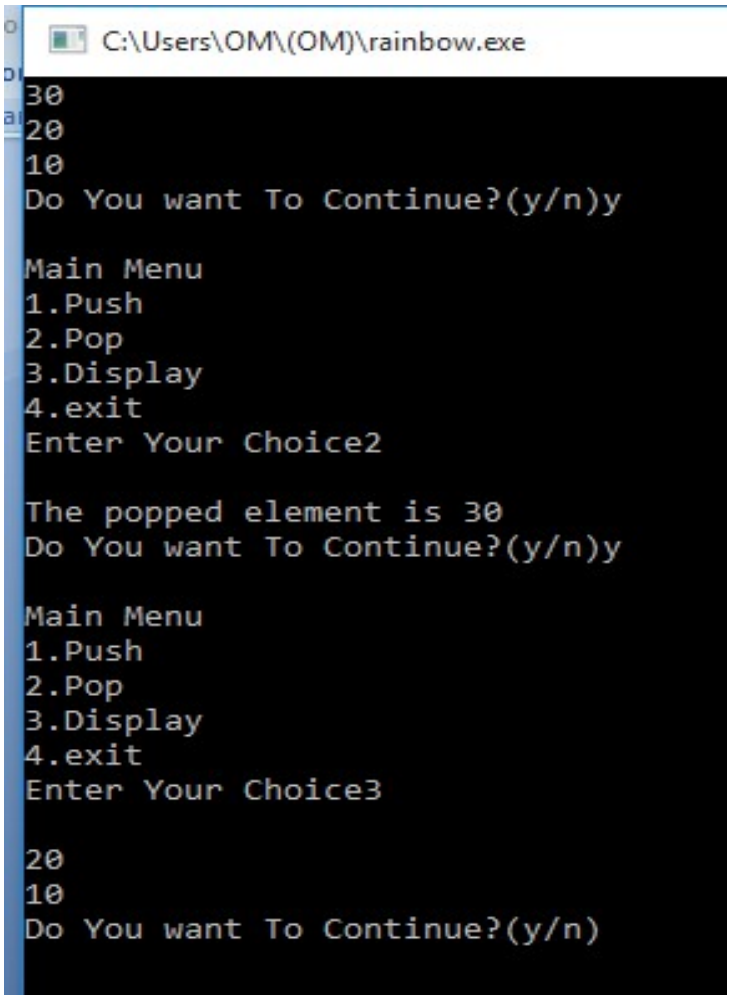


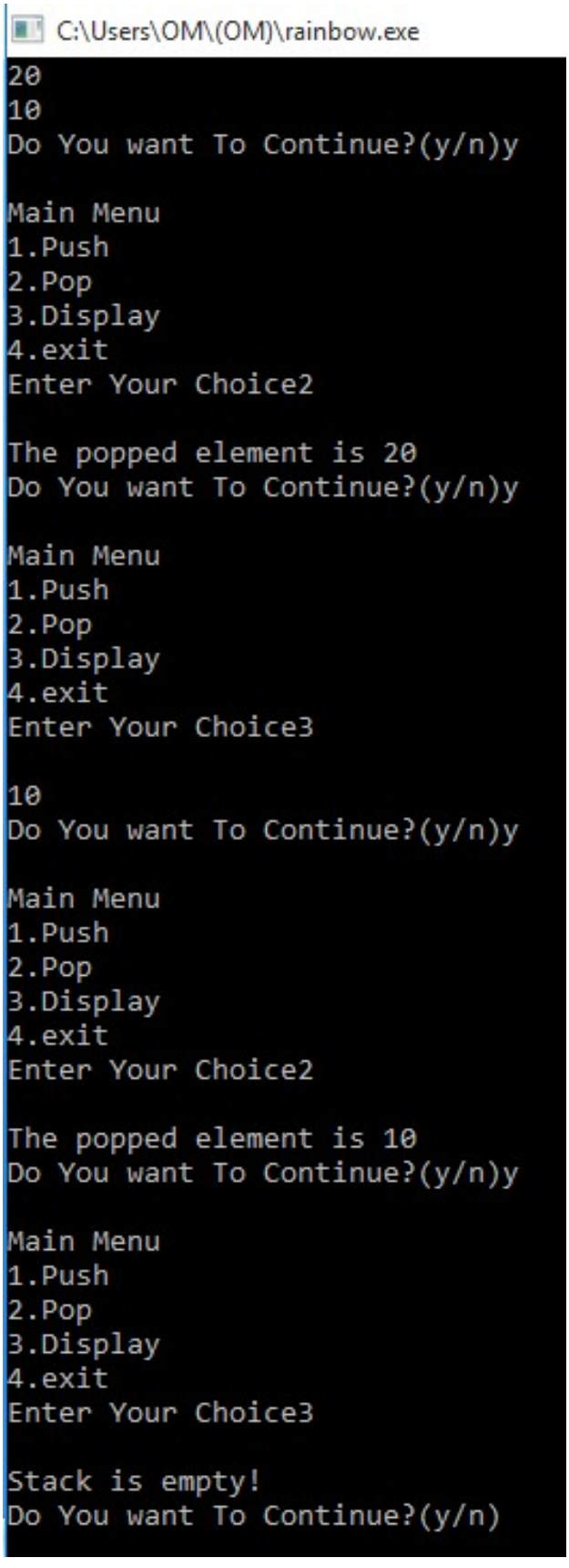
Now we display the elements:




C:\Users\OM\(OM)\rainbow.exe
Do You want To Continue?(y/n)y
Main Menu
1.Push
2.Pop
3.Display
4.exit
Enter Your Choice3
30
20
10
Do You want To Continue?(y/n)

Now we will pop the elements and see the result:




C:\Users\OM\(OM)\rainbow.exe
20
10
Do You want To Continue?(y/n)y
Main Menu
1.Push
2.Pop
3.Display
4.exit
Enter Your Choice2
The popped element is 20
Do You want To Continue?(y/n)y
Main Menu
1.Push
2.Pop
3.Display
4.exit
Enter Your Choice3
10
Do You want To Continue?(y/n)y
Main Menu
1.Push
2.Pop
3.Display
4.exit
Enter Your Choice2
The popped element is 10
Do You want To Continue?(y/n)y
Main Menu
1.Push
2.Pop
3.Display
4.exit
Enter Your Choice3
Stack is empty!
Do You want To Continue?(y/n)

Now we exit from the function: