Name :- Abhinab Roy

Stream :- Computer Science and Engineering

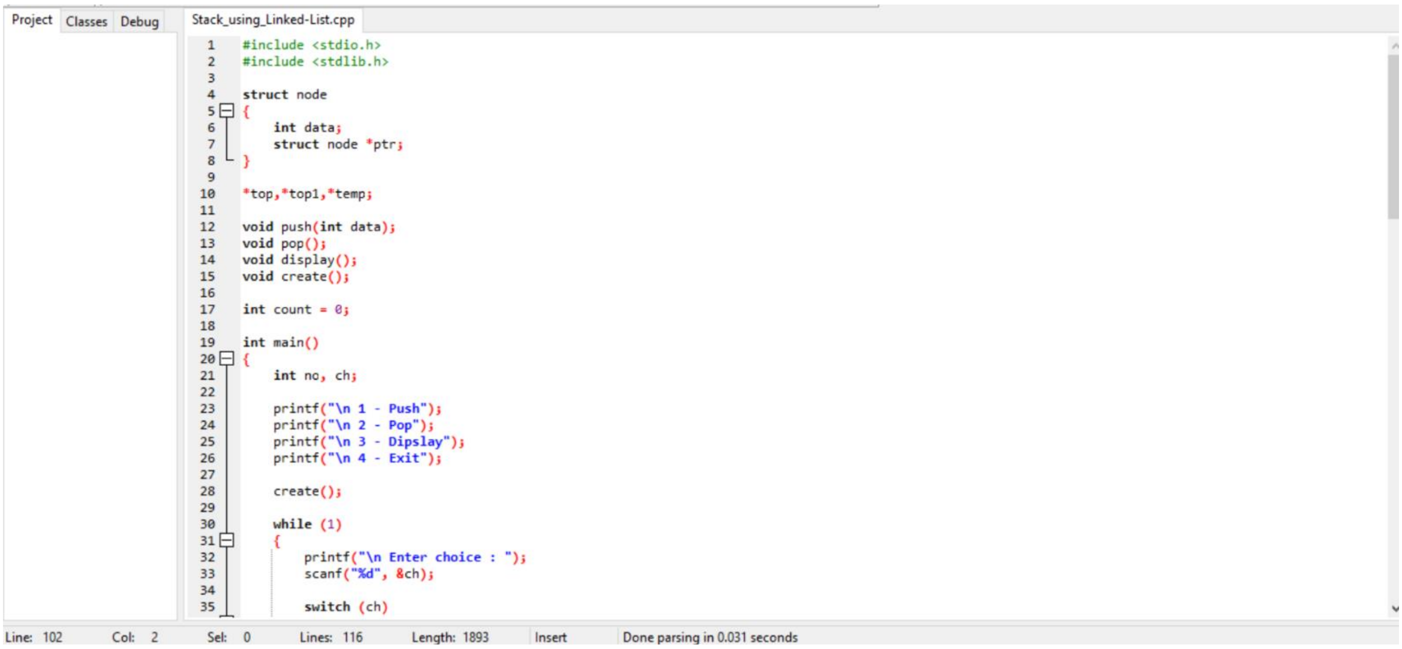
Year :- 2/A

Roll no. :- 26

Uni. Roll no. :- 10900119040

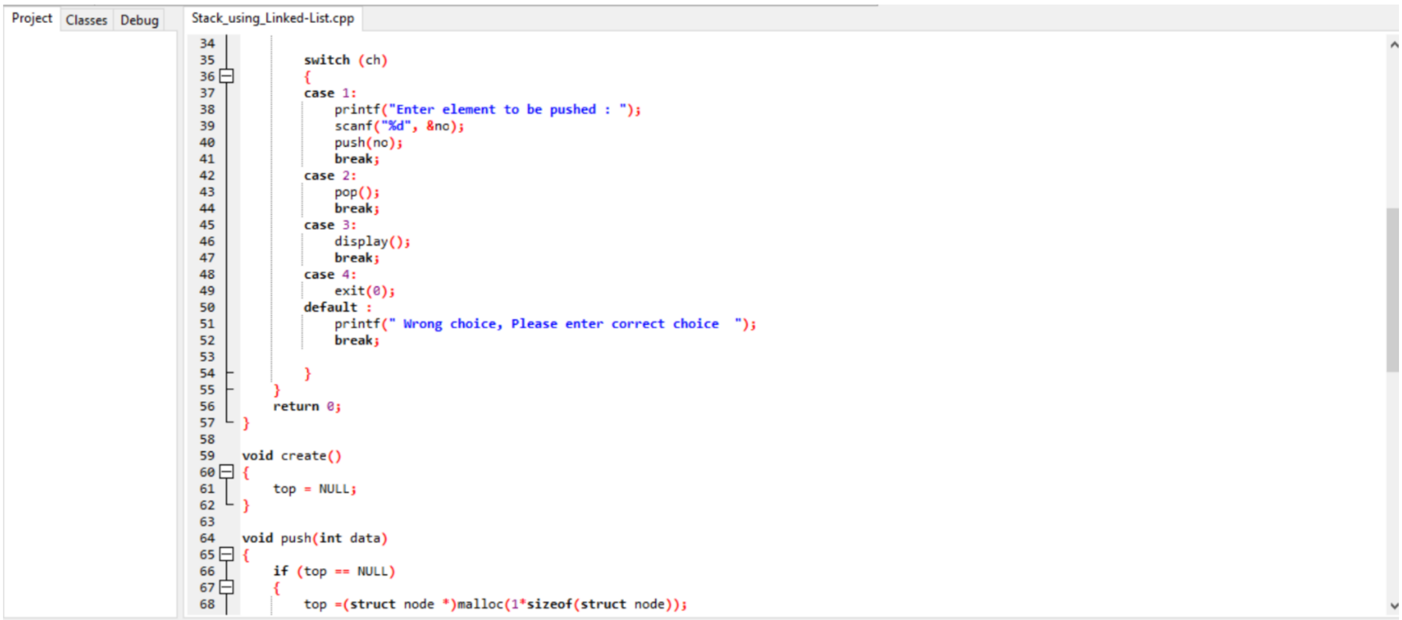
Q) Write a program to implement Stack using Linked List

Code :-

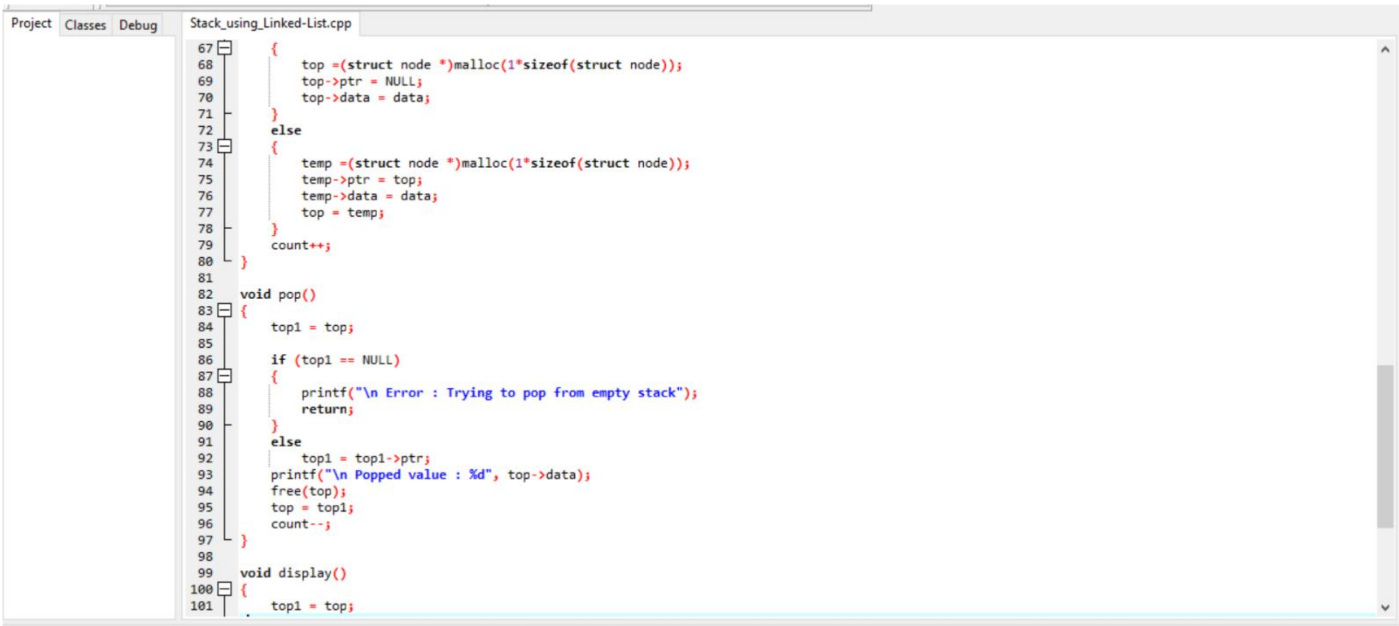


```
#include <stdio.h>
#include <stdlib.h>

struct node
{
    int data;
    struct node *ptr;
}

*top,*top1,*temp;

void push(int data);
void pop();
void display();
void create();

int count = 0;

int main()
{
    int no, ch;

    printf("\n 1 - Push");
    printf("\n 2 - Pop");
    printf("\n 3 - Dipslay");
    printf("\n 4 - Exit");

    create();

    while (1)
    {
        printf("\n Enter choice : ");
        scanf("%d", &ch);

        switch (ch)
```

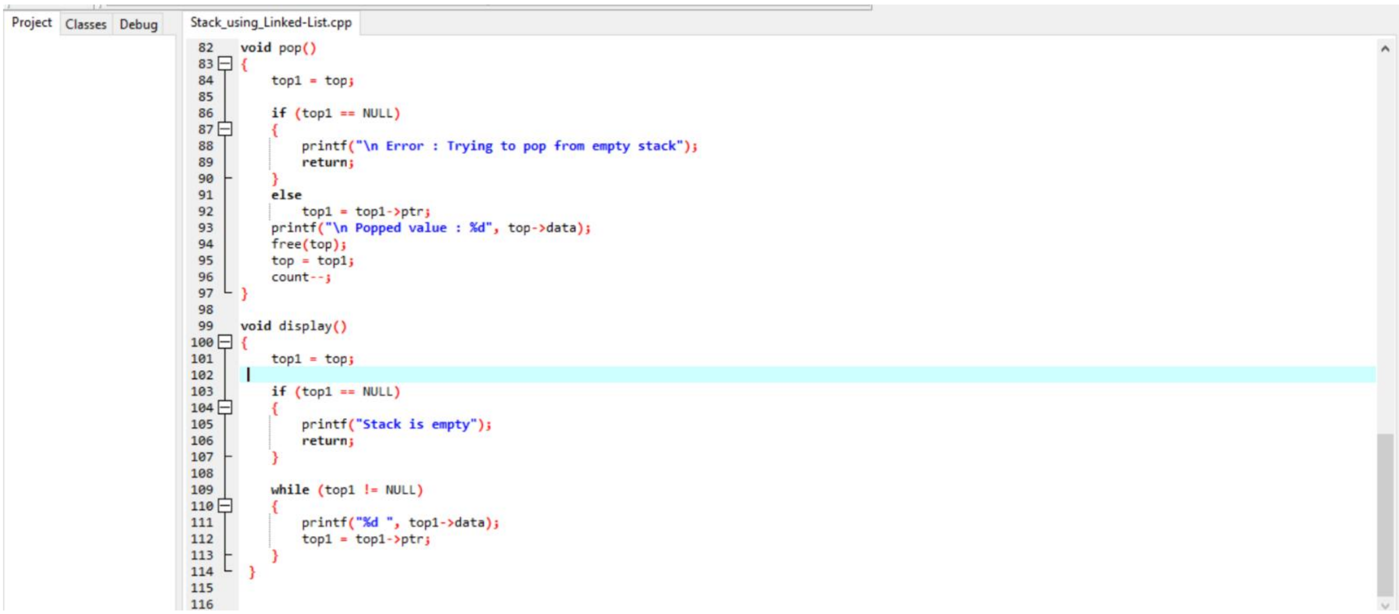
Line: 102 Col: 2 Sel: 0 Lines: 116 Length: 1893 Insert Done parsing in 0.031 seconds



```

        switch (ch)
        {
        case 1:
            printf("Enter element to be pushed : ");
            scanf("%d", &no);
            push(no);
            break;
        case 2:
            pop();
            break;
        case 3:
            display();
            break;
        case 4:
            exit(0);
        default :
            printf(" Wrong choice, Please enter correct choice  ");
            break;

        }
    }
    return 0;
}

void create()
{
    top = NULL;
}

void push(int data)
{
    if (top == NULL)
    {
        top =(struct node *)malloc(1*sizeof(struct node));
```

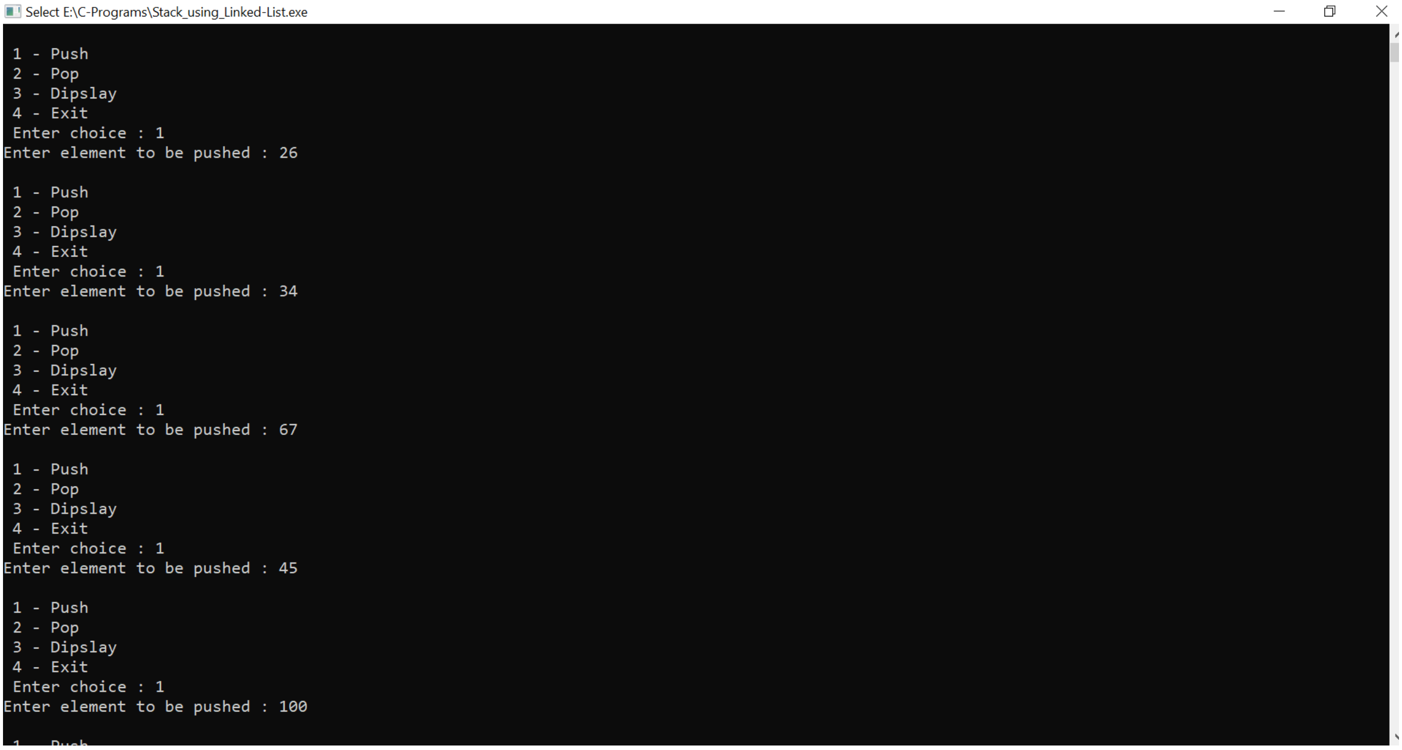


```
    {
        top =(struct node *)malloc(1*sizeof(struct node));
        top->ptr = NULL;
        top->data = data;
    }
    else
    {
        temp =(struct node *)malloc(1*sizeof(struct node));
        temp->ptr = top;
        temp->data = data;
        top = temp;
    }
    count++;
}

void pop()
{
    top1 = top;

    if (top1 == NULL)
    {
        printf("\n Error : Trying to pop from empty stack");
        return;
    }
    else
        top1 = top1->ptr;
    printf("\n Popped value : %d", top->data);
    free(top);
    top = top1;
    count--;
}

void display()
{
    top1 = top;
```

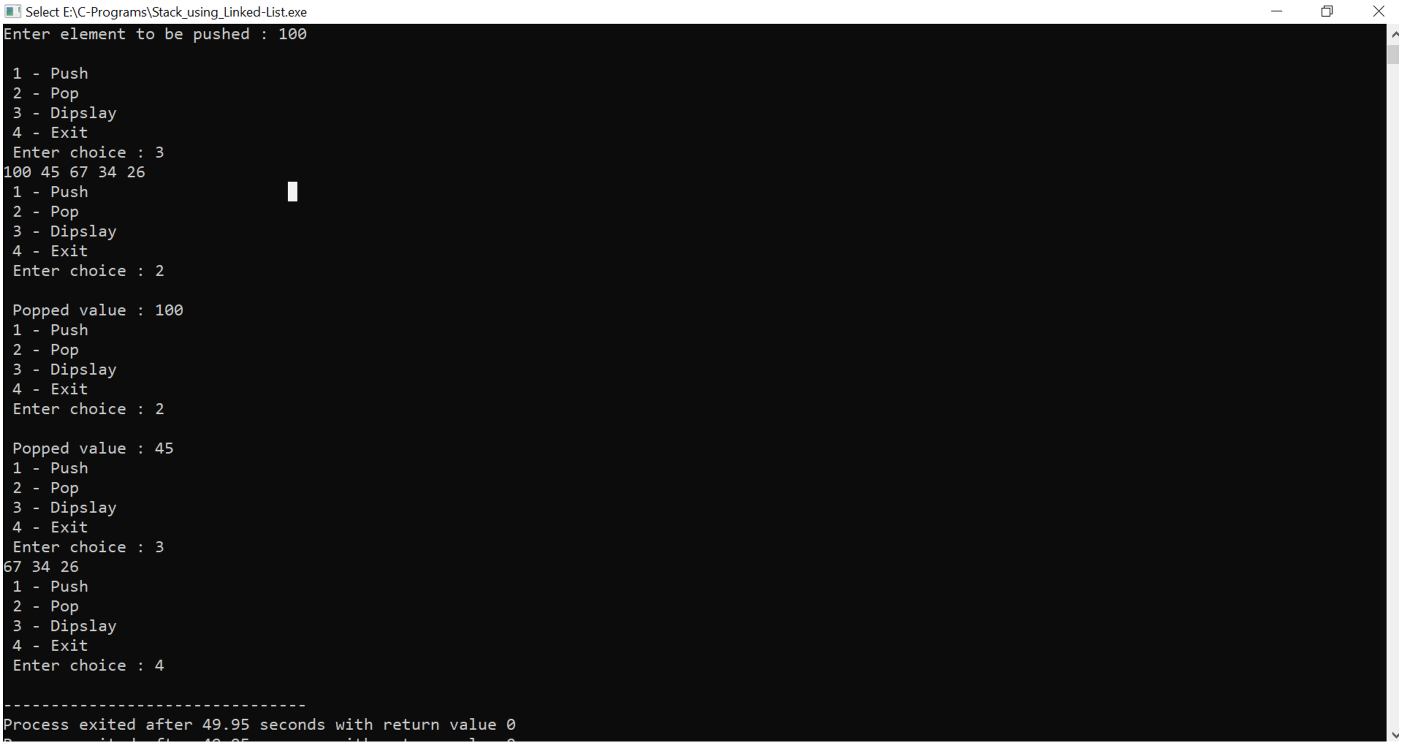


```
void pop()
{
    top1 = top;

    if (top1 == NULL)
    {
        printf("\n Error : Trying to pop from empty stack");
        return;
    }
    else
        top1 = top1->ptr;
    printf("\n Popped value : %d", top->data);
    free(top);
    top = top1;
    count--;
}

void display()
{
    top1 = top;

    if (top1 == NULL)
    {
        printf("Stack is empty");
        return;
    }

    while (top1 != NULL)
    {
        printf("%d ", top1->data);
        top1 = top1->ptr;
    }
 }
```

Output :-



```
 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 1
Enter element to be pushed : 26

 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 1
Enter element to be pushed : 34

 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 1
Enter element to be pushed : 67

 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 1
Enter element to be pushed : 45

 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 1
Enter element to be pushed : 100

 1 - Push
```



```
Enter element to be pushed : 100

 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 3
100 45 67 34 26
 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 2

 Popped value : 100
 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 2

 Popped value : 45
 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 3
67 34 26
 1 - Push
 2 - Pop
 3 - Dipslay
 4 - Exit
 Enter choice : 4

--------------------------------
Process exited after 49.95 seconds with return value 0
Process exited after 49.95 seconds with return value 0
```

END