

Name :- Abhinab Roy

Stream :- Computer Science and Engineering

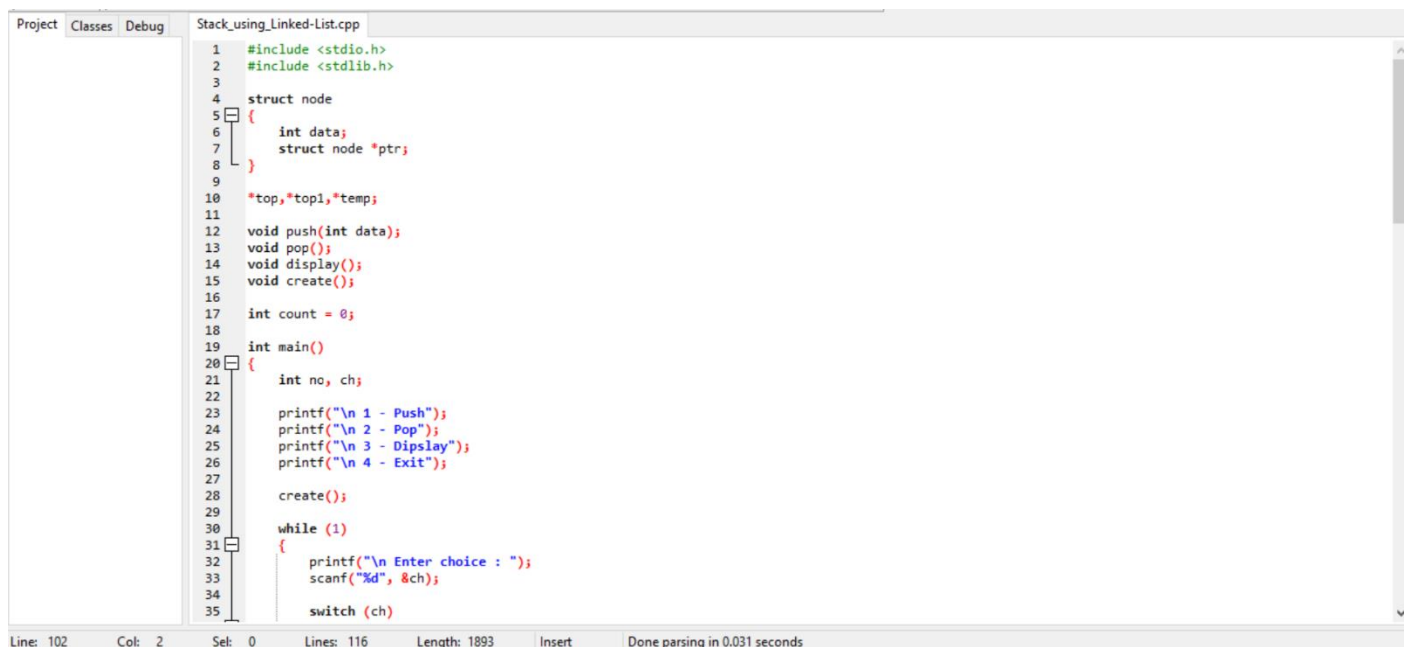
Year :- 2/A

Roll no. :- 26

Uni. Roll no. :- 10900119040

Q) Write a program to implement Stack using Linked List

Code :-



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node
5  {
6      int data;
7      struct node *ptr;
8  }
9
10 *top,*top1,*temp;
11
12 void push(int data);
13 void pop();
14 void display();
15 void create();
16
17 int count = 0;
18
19 int main()
20 {
21     int no, ch;
22
23     printf("\n 1 - Push");
24     printf("\n 2 - Pop");
25     printf("\n 3 - Display");
26     printf("\n 4 - Exit");
27
28     create();
29
30     while (1)
31     {
32         printf("\n Enter choice : ");
33         scanf("%d", &ch);
34
35         switch (ch)
```

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

```
Project Classes Debug Stack_using_Linked-List.cpp
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68

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));
```

```
Project Classes Debug Stack_using_Linked-List.cpp
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

{
    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;
```

```
Project Classes Debug Stack_using_Linked-List.cpp
82 void pop()
83 {
84     top1 = top;
85
86     if (top1 == NULL)
87     {
88         printf("\n Error : Trying to pop from empty stack");
89         return;
90     }
91     else
92     {
93         top1 = top1->ptr;
94         printf("\n Popped value : %d", top->data);
95         free(top);
96         top = top1;
97         count--;
98     }
99
100 void display()
101 {
102     top1 = top;
103
104     if (top1 == NULL)
105     {
106         printf("Stack is empty");
107         return;
108     }
109
110     while (top1 != NULL)
111     {
112         printf("%d ", top1->data);
113         top1 = top1->ptr;
114     }
115 }
116
```

## Output :-

Select E:\C-Programs\Stack\_using\_Linked-List.exe

```
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
```

Select E:\C-Programs\Stack\_using\_Linked-List.exe

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

END