

// 12. Write a C program to implement the application of Stack (Notations)

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

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

```
#define Size 4
```

```
int Top=-1, inp_array[Size];
```

```
void Push();
```

```
void Pop();
```

```
void show();
```

```
int main()
```

```
{
```

```
    int choice;
```

```
    while(1)
```

```
    {
```

```
        printf("\nOperations performed by Stack");
```

```
        printf("\n1.Push the element\n2.Pop the element\n3.Show\n4.End");
```

```
        printf("\n\nEnter the choice:");
```

```
        scanf("%d",&choice);
```

```
        switch(choice)
```

```
        {
```

```
            case 1: Push();
```

```
                    break;
```

```
            case 2: Pop();
```

```
                    break;
```

```
            case 3: show();
```

```
                    break;
```

```
            case 4: exit(0);
```

```
            default: printf("\nInvalid choice!!");
```

```
        }
```

```
    }
```

```
}
```

```
void Push()
```

```
{
```

```
    int x;
```

```
    if(Top==Size-1)
```

```

    {
        printf("\nOverflow!!");
    }
    else
    {
        printf("\nEnter element to be inserted to the stack:");
        scanf("%d",&x);
        Top=Top+1;
        inp_array[Top]=x;
    }
}

```

```

void Pop()
{
    if(Top== -1)
    {
        printf("\nUnderflow!!");
    }
    else
    {
        printf("\nPopped element: %d",inp_array[Top]);
        Top=Top-1;
    }
}

```

```

void show()
{

    if(Top== -1)
    {
        printf("\nUnderflow!!");
    }
    else
    {
        printf("\nElements present in the stack: \n");
        for(int i=Top;i>=0;--i)
            printf("%d\n",inp_array[i]);
    }
}

```

```
D:\data structures lab\stack nota.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug merge sort.c stack notations.c merge.cpp [*] stack nota.cpp
1 // 12. Write a C program to implement the application of Stack (Notations)
2 #include<stdio.h>
3 #include<stdlib.h>
4 #define Size 4
5 int Top=-1, inp_array[Size];
6 void Push();
7 void Pop();
8 void show();
9 int main()
10 {
11     int choice;
12     while(1)
13     {
14         printf("\nOperations performed by Stack");
15         printf("\n1.Push the element\n2.Pop the element\n3.Show\n4.End");
16         printf("\n\nEnter the choice:");
17         scanf("%d",&choice);
18         switch(choice)
19         {
20             case 1: Push();
21                     break;
22             case 2: Pop();
23                     break;
24             case 3: show();
25                     break;
26             case 4: exit(0);
27                     break;
28             default: printf("\nInvalid choice!!");
29                     break;
30         }
31     }
32 }
33 void Push()
34 {
35     int x;
36     if(Top==Size-1)
```

```
D:\data structures lab\stack nota.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug merge sort.c stack notations.c merge.cpp [*] stack nota.cpp
39     else
40     {
41         printf("\nEnter element to be inserted to the stack:");
42         scanf("%d",&x);
43         Top=Top+1;
44         inp_array[Top]=x;
45     }
46 }
47 void Pop()
48 {
49     if(Top== -1)
50     {
51         printf("\nUnderflow!!");
52     }
53     else
54     {
55         printf("\nPopped element: %d",inp_array[Top]);
56         Top=Top-1;
57     }
58 }
59 void show()
60 {
61     if(Top== -1)
62     {
63         printf("\nUnderflow!!");
64     }
65     else
66     {
67         printf("\nElements present in the stack: \n");
68         for(int i=Top;i>=0;i--)
69             printf("%d\n",inp_array[i]);
70     }
71 }
72 }
```

D:\data structures lab\stack nota.exe

Operations performed by Stack

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice:1

Enter element to be inserted to the stack:2

Operations performed by Stack

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice:1

Enter element to be inserted to the stack:3

Operations performed by Stack

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice:1

Enter element to be inserted to the stack:5

Operations performed by Stack

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice:2

Popped element: 5

Operations performed by Stack

1.Push the element

2.Pop the element

3.Show

4.End

Enter the choice:2

Popped element: 3

Operations performed by Stack

1.Push the element

2.Pop the element

3.Show

4.End