



DSA Assignment

Boyapati Sai Venkat

AP19110010174

1st-year CSE-E.

Linked List representation of Stack

1. Write a menu-driven program to perform the following operations (in the form of functions) on a data structure Stack implemented using Linked List.
 - a. Create an empty stack that can accommodate integer
 - b. Insert an element into the stack
 - c. Delete an element from the stack
 - d. Display the content of the stack with an indication of the top element.

Solution:

```
#include <stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *link;
};

void create();

void push();
```

```
void pop();
```

```
void display();
```

```
struct node *Top = NULL;
```

```
void main()
```

```
{
```

```
    int ch;
```

```
    do
```

```
    {
```

```
        printf("\n.....menu driven c program.....");
```

```
        printf("\n 1.Creating a stack\n 2.Inserting an element\n 3.Deleting an element\n 4.Display the contents of element");
```

```
        printf("\n enter your choice");
```

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

```
        switch(ch)
```

```
        {
```

```
            case 1: create();
```

```
                break;
```

```
            case 2: push();
```

```
                break;
```

```
            case 3: pop();
```

```
                break;
```

```
            case 4: display();
```

```
                break;
```

```
            case 5: exit(0);
```

```
        break;
    default:
        printf("Invalid choice entered by the user");
    }
}while(1);
}
```

```
void create()
{
    Top = NULL;
}
```

```
void push()
{
    struct node *temp;
    temp = (struct node*)malloc(sizeof(struct node));
    printf("Enter node data :");
    scanf("%d", &temp->data);
    temp->link = Top;
    Top = temp;
}
```

```
void pop()
{
    struct node *temp;
    if(Top == NULL)
    {
```

```
    printf("no elements to delete");
}
else
{
    temp = Top;
    printf("element: %d", temp->data);

    Top = Top->link;
    temp->link = NULL;
    free(temp);

}
}
```

```
void display()
```

```
{
```

```
    struct node *temp;
```

```
    printf("Top element is ", Top->data);
```

```
    if (Top == NULL)
```

```
    {
```

```
        printf("Stack is empty");
```

```
    }
```

```
    else
```

```
    {
```

```
temp = Top;
while(temp != NULL)
{

    printf("%d\n", temp->data);
    temp = temp->link;

}

}
}
```

Output:

.....menu-driven c program.....

- 1.Creating a stack
 - 2.Inserting an element
 - 3.Deleting an element
 - 4.Display the contents of the element
- enter your choice1

.....menu-driven c program.....

- 1.Creating a stack
- 2.Inserting an element
- 3.Deleting an element



4.Display the contents of the element

enter your choice2

Enter node data:1

.....menu-driven c program.....

1.Creating a stack

2.Inserting an element

3.Deleting an element

4.Display the contents of the element

enter your choice2

Enter node data :3

.....menu-driven c program.....

1.Creating a stack

2.Inserting an element

3.Deleting an element

4.Display the contents of the element

enter your choice2

Enter node data:4

.....menu-driven c program.....

1.Creating a stack

2.Inserting an element

3.Deleting an element

4.Display the contents of the element

enter your choice3

element: 4

.....menu-driven c program.....

- 1.Creating a stack
- 2.Inserting an element
- 3.Deleting an element
- 4.Display the contents of the element

enter your choice44

Invalid choice entered by the user

.....menu-driven c program.....

- 1.Creating a stack
- 2.Inserting an element
- 3.Deleting an element
- 4.Display the contents of the element

enter your choice4

The top element is 3

1

.....menu-driven c program.....

- 1.Creating a stack
- 2.Inserting an element
- 3.Deleting an element
- 4.Display the contents of the element

enter your choice5