	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology **Department of Artificial Intelligence and Data Science**

Name: Siddhesh Dilip Khairnar		
Class: SY	Division: B	Roll No: 272028
Semester: III	Academic Year: 2022-2023	
Subject Name & Code: Data Structure, ADUA21202		
Title of Assignment: Perform implementation of STACK using linked list Push an element on to stack, Pop an element, Display stack		

Assignment No.- 5

PAGE NO.:
DATE: / /

DS Assignment 5

Name : Siddhesh Dilip Khairnar
Roll No: 272028 PRN No: 22110398

Aim: Perform implementation of STACK using linked list Push an element on to stack pop an element Demonstrate overflow simulation on stack Display stack.

Problem statement: Implementation of stack using linked list.

1) Push an element in stack
2) Pop an element
3) Display stack.

Background: Instead of using array, we can also use linked list to implement stack. linked list allocated the memory dynamically. However time complexity in both the scenario are same for all operation that push pop and peek. In linked list implementation of stack, the nodes are maintained non contiguously in memory. Each node contain a pointer to it immediate successor node in the ~~stack~~. st stack. stack is said to be overflow if the space left in the memory heap is not enough to create a node In a stack, Push() is a function used to insert an element into stack. In a stack, Pop() is a function used to delete an element into stack.

Software requirement: online compiler or any IDE

Conclusion: Thus we have successfully implemented stack using linked list and performed operation like push an element on to stack, pop an element, display stack.

Program:

DSA > C++ stack_using_list.cpp > main()

```cpp
#include <iostream>

using namespace std;
struct node // global declaration of structure for node in LL
{
    int data;
    struct node *next;
};
struct node *head = NULL; // global declaration for pointer to head of LL
                          // Dont forget to initialise it to NULL

void push();
void pop();
void display();
```

```
int main()
{
    int choice;

    while (1)
    {

        cout << "\n---------------------------------------";
        cout << "\n1. Push\n2. Pop \n3. Diaplay Stack \n4. Exit";
        cout << "\n Enter your choice:";
        cin >> choice;

        switch (choice)
        {
        case 1:
            cout <<"push";
            push();
            break;

        case 2:
            cout << "\nPop";
            pop();
            break;

        case 3:
            cout << "\nDisplay";
            display();
            break;

        case 4:
            cout << "\nExiting...";
            exit(0);

        default:
            cout << "\n Enter appropriate choice";
        }
    }
}
```

```
void push()
{
    int val;
    struct node *ptr;
    ptr = (struct node *)malloc(sizeof(struct node));
    if (ptr == NULL)
    {
        cout << "Memory value to be pushed";
    }
    else
    {
        cout << "Enter value to be pushed :";
        cin >> val;
        ptr->data = val;
        ptr->next = NULL;
        if (head == NULL)
        {
            head = ptr;
        }
        else
        {
            ptr->next = head;
            head = ptr;
        }
        cout << "The value is pushed onto stack ";
    }
}
```

```
81      void pop(){
82          int val;
83          struct node *ptr;
84          if (head == NULL)
85          {
86              cout<<"stack is empty";
87
88          }
89          else{
90              val=head->data;
91              ptr=head;
92              head=head->next;
93              free(ptr);
94              cout<<endl<<val<<"is popped";
95          }
96
97      }
98      //-----------------------------------------
99      void display(){
100         struct node *ptr;
101         if (head==NULL)
102         {
103             cout<<"the stack is empty:";
104         }
105         else{
106             cout<<"the stack element ";
107             ptr=head;
108             while (ptr!=NULL)
109             {
110                 cout<<endl<<ptr->data;
111                 ptr=ptr->next;
112             }
113
114         }
115     }
```

Output:

```
1. Push
2. Pop
3. Diaplay Stack
4. Exit
 Enter your choice:1
pushEnter value to be pushed :
4
The value is pushed onto stack
----------------------------------------
1. Push
2. Pop
3. Diaplay Stack
4. Exit
```

```
 Enter your choice:2

Pop
4is popped
---------------------------------------
1. Push
2. Pop
3. Diaplay Stack
4. Exit
 Enter your choice:
```

```
4. Exit
 Enter your choice:3

Displaythe stack element
4
---------------------------------------
1. Push
2. Pop
3. Diaplay Stack
4. Exit
 Enter your choice:
```