Ex. No. 3a Singly Linked List

Date:

AIM:

To define a singly linked list node and perform operations such as insertions and deletions dynamically

**Algorithm :**

1. Start

2. Define single linked list node as self referential structure

3. Create Head node with label = -1 and next = NULL using

4. Display menu on list operation

5. Accept user choice

6. If choice = 1 then

Locate node after which insertion is to be done

Create a new node and get data part

Insert new node at appropriate position by manipulating address

Else if choice = 2

Get node's data to be deleted.

Locate the node and delink the node

Rearrange the links

Else

Traverse the list from Head node to node which points to null

7. Stop

**Program:**

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
| **/\* single linked list\*/**  **#include<stdio.h>**  **#include<stdlib.h>**  **struct node**  **{**  **int data;**  **struct node \*next;**  **}\*top,\*new1,\*first;**  **void main()**  **{**  **int wish,opt;**  **void create(),push(),pop(),view();**  **do**  **{**  **system("clear");**  **printf("Stack using linked list menu");**  **printf("\n1.Create\n2.Push\n3.Pop\n4.View\n5.Exit\n");**  **printf("\nEnter your option(1,2,3,4,5):");**  **scanf("%d",&wish);**  **switch(wish)**  **{**  **case 1:**  **create();**  **break;**  **case 2:**  **push();**  **break;**  **case 3:**  **pop();**  **break;**  **case 4:**  **view();**  **break;**  **case 5:**  **exit(0);**  **}**  **printf("\nDo you wnat to continue(0/1):");**  **scanf("%d",&opt);**  **}while(opt==1);**  **}**  **void create()**  **{**  **int ch;**  **top=(struct node\*)malloc(sizeof(struct node));**  **top->next=NULL;**  **do**  **{**  **system("clear");**  **printf("Enter the data:\n");**  **scanf("%d",&top->data);**  **printf("Do you want to insert another(1/0)\n");**  **scanf("%d",&ch);**  **if(ch==1)**  **{**  **new1=(struct node\*)malloc(sizeof(struct node));**  **new1->next=top;**  **top=new1;**  **first=top;**  **}**  **else**  **break;**  **}while(ch==1);**  **}**  **void push()**  **{**  **top=first;**  **new1=(struct node\*)malloc(sizeof(struct node));**  **printf("Enter the element to be pushed:");**  **scanf("%d",&new1->data);**  **new1->next=top;**  **top=new1;**  **first=top;**  **}**  **void pop()**  **{**  **system("clear");**  **top=first;**  **if(top==NULL)**  **printf("\n Stack is empty");**  **else**  **{**  **printf("\nThe element popped out from stack is %d",top->data);**  **top=top->next;**  **first=top;**  **}**  **}**  **void view()**  **{**  **printf("\nStack contents\n");**  **while(top->next!=NULL)**  **{**  **printf("%d->",top->data);**  **top=top->next;**  **}**  **printf("%d\n",top->data);**    **}** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Result :**

Thus operation on single linked list is performed