

EX. NO: 3A

STACK ADT USING LINKED LIST

AIM:

To write a C program for stack ADT using linked list implementation.

DESCRIPTION:

The major problem with the stack implemented using array is, it works only for fixed number of data values. That means the amount of data must be specified at the beginning of the implementation itself. Stack implemented using array is not suitable, when we don't know the size of data which we are going to use. A stack data structure can be implemented by using linked list data structure. The stack implemented using linked list can work for unlimited number of values. That means, stack implemented using linked list works for variable size of data. So, there is no need to fix the size at the beginning of the implementation. The Stack implemented using linked list can organize as many data values as we want.

In linked list implementation of a stack, every new element is inserted as 'top' element. That means every newly inserted element is pointed by 'top'. Whenever we want to remove an element from the stack, simply remove the node which is pointed by 'top' by moving 'top' to its next node in the list. The next field of the first element must be always NULL.

ALGORITHM:

1. Define a struct for each node in the stack. Each node in the stack contains data and link to the next node. TOP pointer points to last node inserted in the stack.
2. The operations on the stack are
 - a. PUSH data into the stack
 - b. POP data out of stack
3. PUSH DATA INTO STACK
 - a. Enter the data to be inserted into stack.
 - b. If TOP is NULL
 - i. The input data is the first node in stack.
 - ii. The link of the node is NULL.
 - iii. TOP points to that node.
 - c. If TOP is NOT NULL
 - i. The link of TOP points to the new node.

ii. TOP points to that node.

4. POP DATA FROM STACK

a. 4a.If TOP is NULL

i. the stack is empty

b. 4b.If TOP is NOT NULL

i. The link of TOP is the current TOP.

ii. The pervious TOP is popped from stack.

5. The stack represented by linked list is traversed to display its content.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<alloc.h> struct
node
{
    int data;    struct
node *next;
}*top,*new1,*first;
void main()
{
    int wish,opt;    void
create(),push(),pop(),view();
    do
    {
        clrscr();
        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 want 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
    {
        clrscr();
        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()
{

    clrscr();    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);

getch();}

```

OUTPUT

```

Stack using linked list menu
1.Create
2.Push
3.Pop
4.View
5.Exit
Enter your option(1,2,3,4,5):1
Enter the data:
23
Do you want to insert another(1/0)
1
Enter the data:
34
Do you want to insert another(1/0)
1
Enter the data:
45
Do you want to insert another(1/0)
1
Enter the data:
54
Do you want to insert another(1/0)
0
Do you want to continue(0/1):1
Stack using linked list menu
1.Create
2.Push
3.Pop

```

```

3.Pop
4.View
5.Exit
Enter your option(1,2,3,4,5):1
Enter the data:
23
Do you want to insert another(1/0)
1
Enter the data:
45
Do you want to insert another(1/0)
0
Do you want to continue(0/1):1
Stack using linked list menu
1.Create
2.Push
3.Pop
4.View
5.Exit
Enter your option(1,2,3,4,5):2
Enter the element to be pushed:40
Do you want to continue(0/1):0
Process returned 0 (0x0)   execution time : 24.456 s
Press any key to continue.

```

RESULT:

Thus the C program for array implementation of Stack ADT was created, executed and output was verified successfully