Expt no: 5

**PROGRAM: Implementation of Stack and Queue Using Linked Lists**

**Input:**

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*link;

};

struct node \*top=NULL;

int main()

{

int choice;

printf("1: push\n2: pop\n3: display\n4: peek\n5:underflow\n6:count\n");

printf("\n0.Exit Program\n");

while(1)

{

printf("\nEnter choice :");

scanf("%d",&choice);

switch(choice)

{

case 0:

printf("\tProgram Exited");

return 0;

break;

case 1:

Push();

break;

case 2:

Pop();

break;

case 3:

Display();

break;

case 4:

Peek();

break;

case 5:

Underflow();

break;

case 6:

Count();

break;

default:

printf("invalid choice\n");

break;

}

}

return 0;

}

void Push()

{

int x;

printf("enter the element to be pushed: ");

scanf("%d",&x);

struct node\*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

if (tmp==NULL)

{

printf("stack Overflow\n");

}

else

tmp->data=x;

tmp->link=top;

top=tmp;

printf("Element %d inserted\n",x);

}

void Pop()

{

struct node\*p;

if(top==NULL)

{

printf("Stack Underflow");

}

else

{

p=top;

top=p->link;

printf("element %d deleted\n",p->data);

free(p);

}

}

void Peek()

{

if (top==NULL)

{printf("stack Underflow");}

printf("Element at the top is %d \n",top->data);

}

void Underflow()

{

if(top==NULL)

{printf("Stack Underflow\n");

return;

}

}

void Display()

{

struct node \*p;

if (top==NULL)

{

printf("List is Empty\n");

return;

}

p=top;

while(p!=NULL)

{

printf("%d\t",p->data);

p=p->link;

}

printf("\n");

}

void Count()

{

struct node \*p;

int count=0;

p=top;

while(p!=NULL)

{

p=p->link;

count++;

}

printf("count = %d \n",count);

}

**Output:**

