**2.Design and implement a stack [linked list] and demonstrate its working with necessary input. Display the appropriate output.**

**ALGORITHM**

**TO PUSH**

Step 1: Start

Step 2: Create a new node with the given data

Step 3: Check whether the stack is empty.

Step 4: if its empty then set the pointer of the node to NULL

Step 5: Else, then make the node point to head

Step 6: Make the new node created as head

Step 7: Stop

**TO POP**

Step 1: Start

Step 2: Check whether the stack is empty

Step 3: If empty, then display “Empty stack”

Step 4: Enter create a temporary node and set it to head

Step 5: Print the data of head

Step 6: Make head to point the next node

Step 7: Delete the temporary node

Step 8: End

**TO DISPLAY**

Step 1: Start

Step 2: Create a temporary node and initialize it with head pointer

Step 3: Check if stack is empty, display “Stack is empty”

Step 4: Else, transfer the temporary node, if NULL is encountered

Step 5: Stop

**PROGRAM**

#include<stdio.h>//standard I/O header file

#include<stdlib.h>//standard library header file

void push();

void pop();

void display();

int main()

{

int choice;//declaring variables

while(1)//checks the condition and enters the while loop

{

printf("Operation performed by stack\n");//printing statement

printf("1.push\n2.Pop\n3.Display\n4.Exit\n");//printing statement

printf("Enter the choice\n");//printing statement

scanf("%d",&choice);//store of value in variable

switch(choice)//switch statement initialization

{

case 1:push();//if option 1 go o push

break;

case 2:pop();if option is 2 go to pop

break;

case 3:display();//if option is 3 go to display

break;

case 4:exit(0);//if option is 4 exit the loop

default:printf("Invalid choice\n");//printing statement

}

}

}

struct node

{

int val;

struct node \*next;

};

struct node \*head;

void push()

{

int val;

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

if(ptr==NULL)//condition check

{

printf("Not able to push the element\n");//printing statement

}

else

{

printf("Enter the value\n");//printing statement

scanf("%d",&val);//store of value in variable

}

if(head==NULL)//condition check

{

ptr->val=val;

ptr->next=NULL;

head=ptr;

}

else

{

{

ptr->val=val;

ptr->next=head;

head=ptr;

}

printf("Item pushed\n");//printing statement

}

}

void pop()

{

int item;

struct node \*ptr;

if(head==NULL)//condition check

{

printf("Underflow\n");//printing statement

}

else

{

item=head->val;

ptr=head;

head=head->next;

free(ptr);

printf("Item popped\n");//printing statement

}

}

void display()

{

int i;

struct node \*ptr;

ptr=head;

if(ptr==NULL)//condition check

{

printf("Stack is empty\n");//printing statement

}

else

{

printf("printing stack\n");//printing statement

while(ptr!=NULL)//condition check

{

printf("%d\n",ptr->val);//printing elements

ptr=ptr->next;

}

}

}

**OUTPUT**



