AIM:-Design and implementation of a stack and demonstrate its working with necessary input. Display the appropriate output

Alg0ritm

1)push ()

Step 1: Start.

Step 2: Create a new node with the given data.

Step 3: Check whether the stack is empty (head==null).

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.

2)pop ()

Step 1: Start.

Step 2: Check whether is empty.

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

Step 4: Enter create a temporary node and set it to the 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: Stop.

3)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, transverse the temporary node. It null is encountered.

Step 5: stop.

PROGRAM:

#include<stdio.h>//standard input output header files//

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

void push();//push function//

void pop();//pop function//f

void display();// display function//

int main()//main function//

{

int choice;//declaring choice//

while(1)

{

printf("operation performed by stack\n");//prints the statement//

printf("1.push\n");//prints the functions//

printf("2.pop\n");//prints the function//

printf("3.display\n");//prints the function//

printf("4.exit\n");//prints the functions//

printf("enter the choice\n")//print the statement//

scanf("%d",&choice);//scans the input and place into choice//

switch(choice) //sacns the condition and enters into the case when it satisfies//

{

case 1:push();//prints the case 1 when the condition is 1//

break;

case 2:pop();// prints the case 2 when the conditon is 2//

break;

case 3:display();// prints the case 3 when the condition is 3//

break;

case 4:exit(0); // come out of the loop//

default : printf("invaild choice|n");//prints the value//

}

}

}

struct node//node decleration//

{

int val;

struct node \*next;

};

struct node \*head;

void push()

{

int val;//declaring the x//

struct node \*ptr+(struct node \*)malloc(sizeof(struct node));//dynamic memory allocation od malloc function//

if(ptr==NULL)//checks the condition//

{

printf("not able to push the element \n");//prints the statement//

scanf("%d",&val);

}

else

{

printf("enter the value\n");//prints the statement//

scanf("%d",&val);

}

if (head==NULL)//checks the condition//

{

ptr->val+val;//pointer value//

ptr->next=NULL;

head=ptr;

}

else

{

ptr->val=val;

ptr->next=head;

head+ptr;

}

printf("item pushed \n");

}

void pop()

{

int item:

struct node \*ptr;

if (head ==NULL)

{

Printf("under flow\n");

}

else

{

item=head->val;

ptr=head->next;

free(ptr);

printf("item popped\n");

}

}

void display()

{

inti;//declaration of i//

struct node \*ptr;

ptr=head;

if(ptr==NULL)

{

printf("stack is empty\n");

}

else

{

prinft("printing stack\n");

while(ptr!=NULL);

{

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

ptr=ptr->next;

}

}

}

Screeshot:

