**ASHESH KUMAR**

**501254**

**I.T. – III**

**DYNAMIC IMPLEMENTATION OF STACK**

#include<stdio.h>

#include<stdlib.h>

void push();

void pop();

void display();

struct node {

int info;

struct node \*link;

} \*start = NULL, \*new, \*temp, \*p;

int main() {

char choice;

int a;

do {

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("A.PUSH\nB.POP\nD.DISPLAY\nE.EXIT\n");

printf("GIVE CHOICE: ");

choice = getchar();

fflush(stdin); //clear any extra characters or whitespaces typed in after "choice"

printf("\n");

switch (choice) {

case 'A':

case 'a':

push();

break;

case 'B':

case 'b':

pop();

break;

case 'D':

case 'd':

display();

break;

case 'E':

case 'e':

printf("PROGRAM WILL EXIT\_\_");

exit(0); //exits all loops

default:

printf("INCORRECT CHOICE! TRY AGAIN!!\n\n");

}

} while (choice != 'e' || choice != 'E');

return (0);

}

void push() {

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

printf("\nEnter the item : ");

scanf("%d", &new->info);

fflush(stdin);

new->link = NULL;

/\*If stack is empty\*/

if (start == NULL)

start = new;

/\* Otherwise move to end(top) of the stack. \*/

else {

p = start;

while (p->link != NULL)

p = p->link;

p->link = new;

}

printf("%d PUSHED!", new->info);

}

void pop() {

/\*If stack is empty\*/

if (start == NULL)

printf("\nOPERATION FAILED.\nTHE STACK IS EMPTY!!\n\n");

/\*If there is only one item\*/

else if (start->link == NULL) {

printf("\nThe deleted element is : %d", start->info);

free(start);

start = NULL;

}/\*Else, move to last element\*/

else {

/\*'p' holds last(top) element and 'temp' holds second last element\*/

p = start;

while (p->link != NULL) {

temp = p;

p = p->link;

}

printf("\nDeleted element is : %d\n", p->info);

temp->link = NULL;

free(p);

}

}

void display() {

if (start == NULL)

printf("\nStack is empty");

else {

printf("\nThe elements are : ");

p = start;

while (p != NULL) {

printf("%d", p->info);

printf("\t");

p = p->link;

}

printf("<-TOP \n");

}

}

**OUTPUT**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A.PUSH

B.POP

D.DISPLAY

E.EXIT

GIVE CHOICE: a

Enter the item : 23

23 PUSHED!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A.PUSH

B.POP

D.DISPLAY

E.EXIT

GIVE CHOICE: A

Enter the item : -98

-98 PUSHED!

.

.

.

.

.

.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A.PUSH

B.POP

D.DISPLAY

E.EXIT

GIVE CHOICE: d

The elements are : 23 54 78 0 76 78 21 -98 <-TOP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A.PUSH

B.POP

D.DISPLAY

E.EXIT

GIVE CHOICE: b

Deleted element is : -98

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A.PUSH

B.POP

D.DISPLAY

E.EXIT

GIVE CHOICE: d

The elements are : 23 54 78 0 76 78 21 <-TOP

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A.PUSH

B.POP

D.DISPLAY

E.EXIT

GIVE CHOICE:e

PROGRAM WILL EXIT\_\_