//Stack using linked list

#include<stdio.h>

#include<conio.h>

#include<malloc.h>

#include<stdlib.h>

#define SIZE 5

struct stack

{

int data;

struct stack \*link;

};

struct stack \*TOP=NULL, \*HEAD=NULL;

int count = 0;

void push()

{

struct stack \*temp;

//To count the nodes in the stack

temp = HEAD;

while(temp!=NULL)

{

count = count +1;

temp = temp->link;

}

if(count == SIZE) //logical overflow condition

{

printf("\nStack overflow!can't push ");

}

else

{

//New node creation

temp=(struct stack \*)malloc(sizeof(struct stack));

if(temp==NULL)

{

printf("\nMemory is full");

getch();

return;

}

//Read value to insert

printf("\nEnter item to push:");

scanf("%d",&temp->data);

temp->link= NULL;

//updation of TOP after insertion

if(TOP == NULL) //Empty queue

{

HEAD = temp;

TOP = temp;

}

else // At least one node is there in the queue

{

TOP->link= temp;

TOP = TOP->link;

}

}

}

void del\_stack()

{

struct stack \*temp1,\*temp2;

if(TOP == NULL)

{

printf("\nStack underflow!no item to delete");

}

else

{

if(HEAD->link==NULL)

{

printf("\nDeleted item is %d ",TOP->data);

free(TOP);

HEAD = NULL;

TOP = NULL;

}

else

{

temp2=HEAD;

while(temp2->link!=NULL)

{

temp1=temp2;

temp2=temp2->link;

}

printf("\nDeleted item is %d ",TOP->data);

free(TOP);

TOP=temp1;

TOP->link= NULL;

}

}

}

void display()

{

struct stack \*temp;

if(TOP == NULL)

{

printf("\nStack underflow!no item to delete");

}

else

{

printf("\nThe items in the stack are\n");

temp=HEAD;

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->link;

}

}

}

void main()

{

int ch;

clrscr();

while(1)

{

printf("\n1. Push\n2. Pop\n3. Display\n4. Exit\n");

printf("Enter your choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1: push();

break;

case 2: del\_stack();

break;

case 3: display();

break;

case 4: exit(0);

default: printf("\nWrong choice: ");

}

}

}