**DSA PRACTICAL FILE**

(CodeChef & online GDB as C language IDE is used for all the programs)

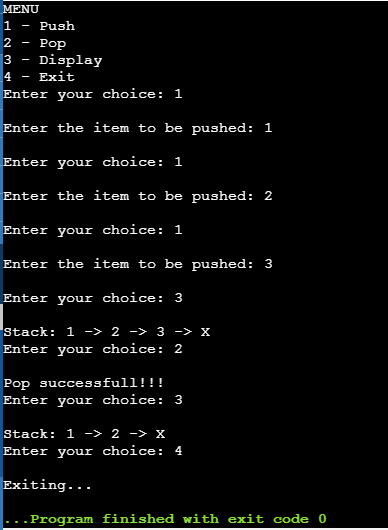
Practical File programs from 13 to 16

Name: Hardik Chhabra

ECAM-1

Roll no.: 2020UEA6504

**OUTPUT-13**



**Practical-13**

**Q. Write a program to implement a stack using an array.**

#include <stdio.h>

int A[10],top=-1,n;

void push() {

if(top==9) {

printf("\nOVERFLOW!!!");

}else {

top++;

printf("\nEnter the item to be pushed: ");

scanf("%d",&A[top]);

}

}

void pop() {

if(top==-1) {

printf("\nUNDERFLOW!!!");

}else {

printf("\nPop successfull!!!");

top--;

}

}

void display() {

if(top==-1) {

printf("\nUNDERFLOW!!!");

}else {

printf("\nStack:");

for (int i = 0; i <= top; i++) {

printf(" %d ->",A[i]);

}

printf(" X");

}

}

int main(void) {

// your code goes here

int x=1,choice;

printf("MENU\n1 - Push\n2 - Pop\n3 - Display\n4 - Exit");

while(x) {

printf("\nEnter your choice: ");

scanf("%d",&choice);

if(choice==1) {push();}

else if(choice==2) {pop();}

else if(choice==3) {display();}

else if(choice==4) {x=0; printf("\nExiting...");}

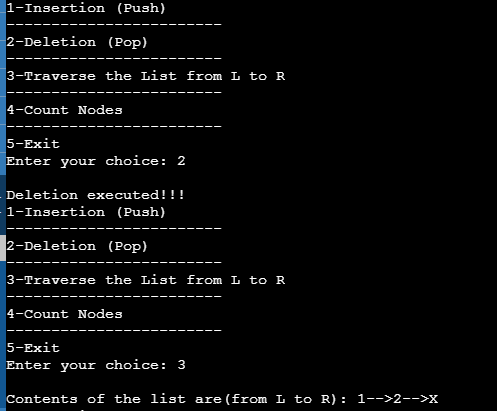
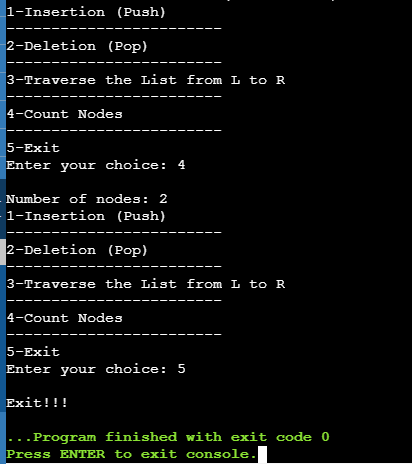
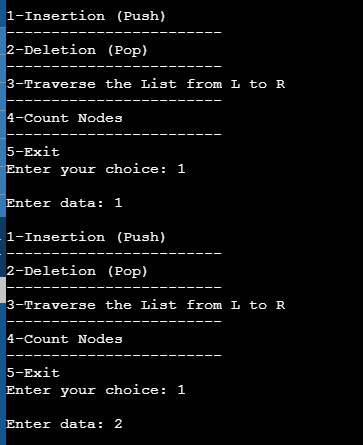
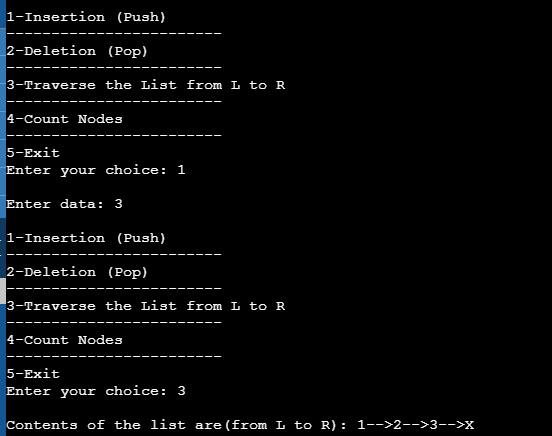
else {printf("Enter a valid choice!!!");}

}

return 0;

}

**OUTPUT-14**



**Practical-14**

**Q. Write a program to implement a stack using a linked list.**

#include<stdio.h>

#include<stdlib.h>

struct stack {

int data;

struct stack \*next;

};

typedef struct stack node;

node \*start=NULL;

node\* getnode() {

node\* newnode;

newnode=(node\*)malloc(sizeof(node));

printf("\nEnter data: ");

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

newnode->next=NULL;

return newnode;

}

int menu() {

int ch;

printf("\n1-Insertion (Push)");

printf("\n------------------------");

printf("\n2-Deletion (Pop)");

printf("\n------------------------");

printf("\n3-Traverse the List from L to R");

printf("\n------------------------");

printf("\n4-Count Nodes");

printf("\n------------------------");

printf("\n5-Exit");

printf("\nEnter your choice: ");

scanf("%d",&ch);

return ch;

}

void createLL(int n) {

node \*temp, \*newnode;

for (int i = 0; i < n; i++) {

newnode=getnode();

if(start==NULL) {start=newnode;}

else {

temp=start;

while(temp->next!=NULL) {temp=temp->next;}

temp->next=newnode;

}

}

}

int countnode(node \*ptr) {

int ctr=0;

while(ptr!=NULL) {

ctr++;

ptr=ptr->next;

}

return ctr;

}

void traverse() {

node \*temp;

temp=start;

if(start==NULL) {printf("\nList is Empty!!!");}

else {

printf("\nContents of the list are(from L to R): ");

while(temp!=NULL) {

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

temp=temp->next;

}

}

printf("X");

}

void push() {

node \*newnode,\*temp;

newnode=getnode();

if(start==NULL) {start=newnode;}

else {

temp=start;

while(temp->next!=NULL) {temp=temp->next;}

temp->next=newnode;

}

}

void pop() {

node \*temp, \*prev;

if(start==NULL) {

printf("\nNo nodes exist!!!");

return;

}else {

temp=start;

prev=start;

while(temp->next!=NULL) {

prev=temp;

temp=temp->next;

}

prev->next=NULL;

free(temp);

printf("\nDeletion executed!!!");

}

}

int main(void) {

int ch,n;

while(1) {

ch=menu();

switch(ch) {

case 1: push();

break;

case 2: pop();

break;

case 3: traverse();

break;

case 4: printf("\nNumber of nodes: %d",countnode(start));

break;

case 5: printf("\nExit!!!");

exit(0);

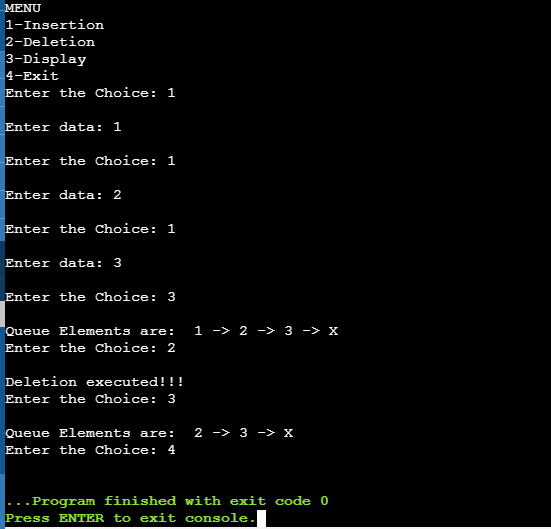
}

}

return 0;

}

**OUTPUT-15**



**Practical-15**

**Q. Write a program to implement a queue using an array.**

#include<stdio.h>

#define n 10

int main(void) {

int queue[n],ch=1,front=0,rear=0,i,j=1,x=n;

printf("MENU");

printf("\n1-Insertion");

printf("\n2-Deletion");

printf("\n3-Display");

printf("\n4-Exit");

while(1) {

printf("\nEnter the Choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1:

if(rear==x) {

printf("\nQueue is Full");

}else {

printf("\nEnter data: ");

scanf("%d",&queue[rear++]);

}

break;

case 2:

if(front==rear) {

printf("\nQueue is empty");

}else {

printf("\nDeletion executed!!!");

front++;

x++;

}

break;

case 3:

printf("\nQueue Elements are: ");

if(front==rear) {

printf("\nQueue is Empty");

}else {

for(i=front; i<rear; i++) {

printf(" %d ->",queue[i]);

}

printf(" X");

break;

case 4:

exit(0);

default:

printf("\nWrong Choice!!!");

}

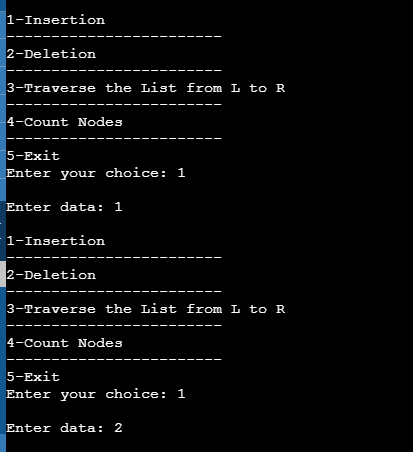
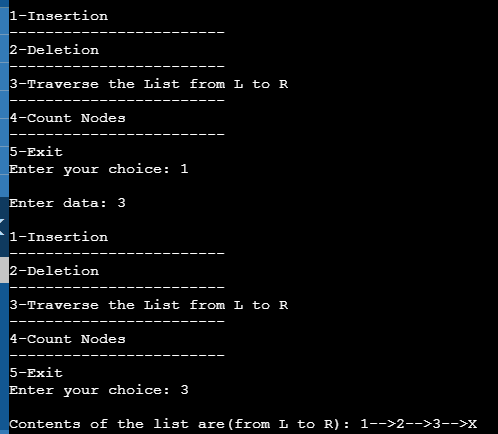
}

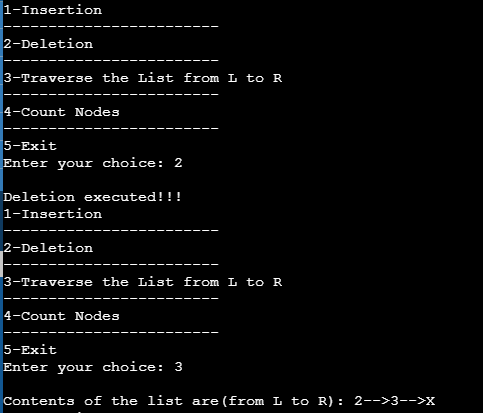
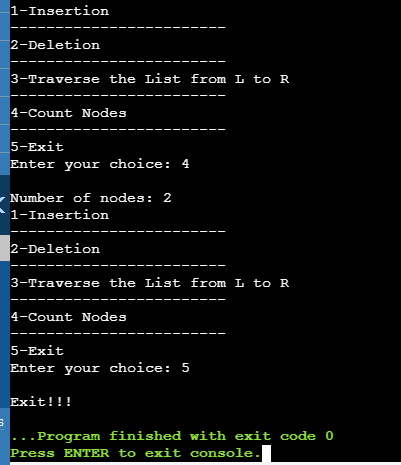
}

return 0;

}

**OUTPUT-16**

****

****

**Practical-16**

**Q. Write a program to implement a queue using a linked list.**

#include<stdio.h>

#include<stdlib.h>

struct queue {

int data;

struct queue \*next;

};

typedef struct queue node;

node \*start=NULL;

node\* getnode() {

node\* newnode;

newnode=(node\*)malloc(sizeof(node));

printf("\nEnter data: ");

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

newnode->next=NULL;

return newnode;

}

int menu() {

int ch;

printf("\n1-Insertion");

printf("\n------------------------");

printf("\n2-Deletion");

printf("\n------------------------");

printf("\n3-Traverse the List from L to R");

printf("\n------------------------");

printf("\n4-Count Nodes");

printf("\n------------------------");

printf("\n5-Exit");

printf("\nEnter your choice: ");

scanf("%d",&ch);

return ch;

}

int countnode(node \*ptr) {

int ctr=0;

while(ptr!=NULL) {

ctr++;

ptr=ptr->next;

}

return ctr;

}

void traverse() {

node \*temp;

temp=start;

if(start==NULL) {printf("\nList is Empty!!!");}

else {

printf("\nContents of the list are(from L to R): ");

while(temp!=NULL) {

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

temp=temp->next;

}

}

printf("X");

}

void ins() {

node \*newnode,\*temp;

newnode=getnode();

if(start==NULL) {start=newnode;}

else {

temp=start;

while(temp->next!=NULL) {temp=temp->next;}

temp->next=newnode;

}

}

void del() {

node \*temp;

if(start==NULL) {

printf("\nNo nodes exist!!!");

return;

}else {

temp=start;

start=temp->next;

free(temp);

printf("\nDeletion executed!!!");

}

}

int main(void) {

int ch,n;

while(1) {

ch=menu();

switch(ch) {

case 1: ins();

break;

case 2: del();

break;

case 3: traverse();

break;

case 4: printf("\nNumber of nodes: %d",countnode(start));

break;

case 5: printf("\nExit!!!");

exit(0);

}

}

return 0;

}