**Date:15.09.22**

**Program 6: To perform stack operations**

1. **Push**
2. **Pop**

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define max 5

int a[max],top=-1;

void push(int n)

{

if(top==max-1)

printf("\n OverFlow");

else

{

top++;

a[top]=n;

}

}

void pop()

{

int del;

if(top==-1)

printf("\n Underflow \t");

else

{

del=a[top];

printf("\n Deleting \t %d",del);

top--;

}

}

void Displaying\_elements()

{

int i;

if(top==-1)

puts("stack is empty");

else

{

for(i=top;i>=0;i--)

printf("\t%d",a[i]);

}

}

int main()

{

int choice,n;

do

{

printf("\n 1.Push");

printf("\n 2.Pop");

printf("\n 3.Display");

printf("\n Enter Choice \t");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("enter the element to push:");

scanf("%d",&n);

push(n);

break;

case 2:

pop();

break;

case 3:

Displaying\_elements();

break;

case 4:

exit(0);

break;

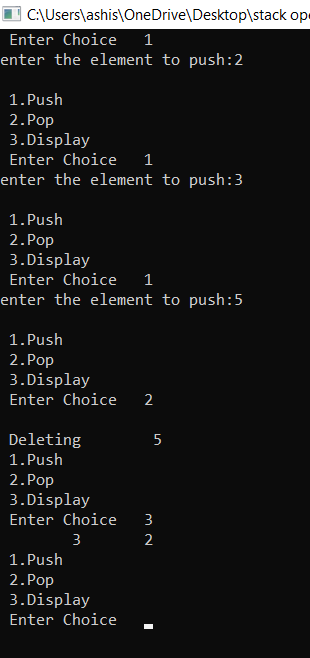
}

}while(1);

return 0;

}

**Output:**

****

**Program 7: To perform Tower of Hanoi using stack operations**

**Code:**

#include<stdio.h>

void TOHanoi(int n,char x,char y,char z)

{

if(n==1)

printf("\n%c to %c", x,z);

if(n>1)

{

TOHanoi(n-1,x,z,y);

printf("\n%c to %c",x,z);

TOHanoi(n-1,z,y,x);

}

}

int main()

{

int n;

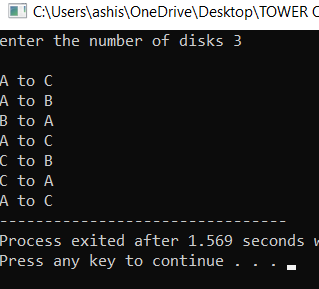
printf("enter the number of disks ");

scanf("%d",&n);

TOHanoi(n,'A','B','C');

}

**Output:**

****

**Program 8: To perform queue operations**

1. **Rear (Insertion)**
2. **Front (Deletion)**

**Code:**

#include<stdio.h>

#include<stdlib.h>

#define SIZE 5

int front=-1;

int rear=-1;

int a[SIZE];

void rear\_of\_queue();

void front\_of\_queue();

void display();

int main()

{

int choice;

do

{

printf("\n 1. Insert");

printf("\n 2. Delete");

printf("\n 3. Display ");

printf("\n 4. Exit");

printf("\n Enter Your Choice:");

scanf("%d",&choice);

switch(choice)

{

case 1:

rear\_of\_queue();

break;

case 2:

front\_of\_queue();

break;

case 3:

display();

break;

case 4:

printf("wrong choice");

exit(0);

}

}while(choice!=4);

}

void rear\_of\_queue()

{

int no;

printf("\n Enter No.:");

scanf("%d",&no);

if(rear < SIZE-1)

{

a[++rear]=no;

if(front==-1)

front=0;

}

else

{

printf("\n Queue overflow");

}

}

void front\_of\_queue()

{

if(front==-1)

{

printf("\nQueue Underflow");

return;

}

else

{

printf("\nDeleted Item:%d\n",a[front]);

}

if(front==rear)

{

front=-1;

rear=-1;

}

else

{

front=front+1;

}

}

void display()

{

int i;

if(front==-1)

{

printf("\nQueue is empty....");

return;

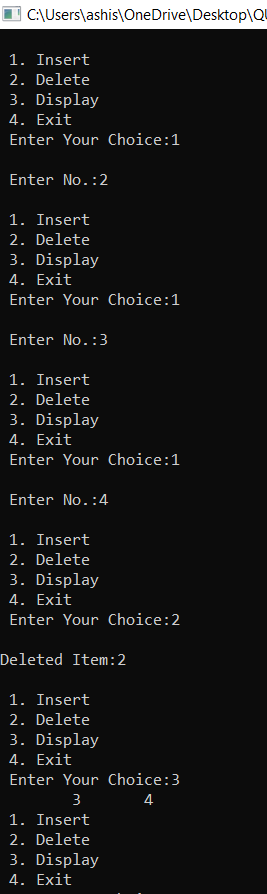
}

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

printf("\t%d",a[i]);

}

**Output:**

****