NAME: BHOOMIKA.M

USN: 1BM23CS068

LAB-02

INFIX TO POSTFIX

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#define size 500

char s[size];

int top=-1;

void push(char x){

if(top<size-1){

top++;

s[top]=x;

}

}

char pop(){

if(top>0){

return s[top--];}

}

int pre(char sys){

if(sys=='^')return 3;

else if(sys=='\*'||sys=='/')return 2;

else if(sys=='+'||sys=='-')return 1;

else return 0;

}

void main(){

char infix[500],ch,postfix[500],ele;

int i=0,k=0;

printf("enter the expression");

scanf("%s",infix);

push('#');

while((ch=infix[i++])!='\0'){

if(isalnum(ch))postfix[k++]=ch;

else if(ch=='(')push(ch);

else if(ch==')'){

while(top>0 && s[top]!='('){

postfix[k++]=pop();

}

ele=pop();

}

else{

while(top>0 && pre(s[top])>=pre(ch)){

postfix[k++]=pop();

}

push(ch);

}

}

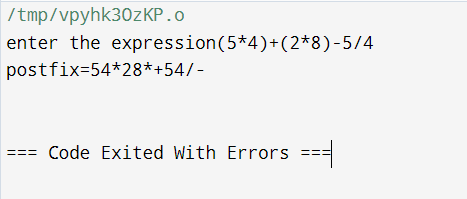
while(top>0 && s[top]!='#')

postfix[k++]=pop();

postfix[k]='\0';

printf("postfix=%s\n",postfix);

}



LAB-03

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#define size 100

int q[size],front=-1,rear=-1;

void enqueue(){

int x;

printf("enter a number");

scanf("%d",&x);

if (rear==(size-1))

printf("queue is full");

else {

if (rear==-1){

front=rear=0;

}

else {

rear++;

}

q[rear]=x;

}

}

void dequeue(){

if(rear==-1)printf("queue is empty");

else{

printf("%d",q[front]);

front++;

}

}

void display(){

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

printf("%d\n",q[i]);

}

void main(){

int n;

printf("enter choice 1.enqueue 2. dequeue 3. display 4. exit");

while(1){

scanf("%d",&n);

switch(n){

case 1:

enqueue();

break;

case 2: dequeue();

break;

case 3: display();

break;

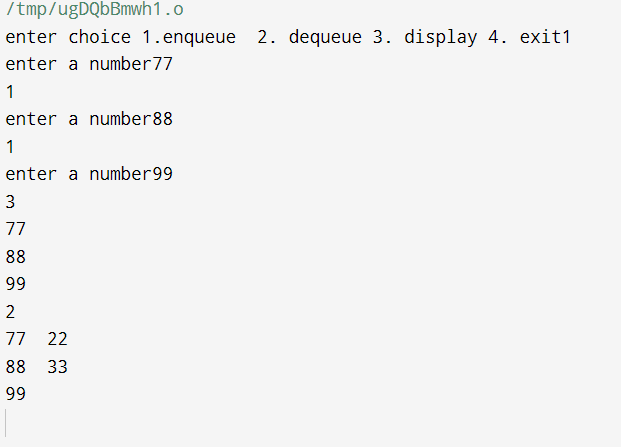
case 4: exit(0);

default: printf("invalid");

}

}

}



2. CIRCULAR QUEUE

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#define size 5

int q[size],front=-1,rear=-1;

void enqueue(){

int x;

printf("enter a number");

scanf("%d",&x);

if ((front==rear+1) || (front==0 && rear==-1))

printf("queue is full");

else {

if (rear==-1){

front=rear=0;

}

else {

rear=(1+rear)%size;

}

q[rear]=x;

}

}

void dequeue(){

if(rear==-1)printf("queue is empty");

else{

printf("%d",q[front]);

if(front==rear){

front=-1;

rear=-1;

}

else

front=(1+front)%size;

}

}

void display(){

if (front==-1)printf("empty");

else{

for(int i=front;i!=rear;i=(i+1)%size)

printf("%d\n",q[i]);

printf("%d",q[rear]);

}

}

void main(){

int n;

printf("enter choice 1.enqueue 2. dequeue 3. display 4. exit");

while(1){

scanf("%d",&n);

switch(n){

case 1:

enqueue();

break;

case 2: dequeue();

break;

case 3: display();

break;

case 4: exit(0);

default: printf("invalid");

}

}

}

