*Digital Assignment - 1*

***Code 1: Queue Operations***

#include<stdio.h>

float ar[10];

int f = -1;

int r = -1;

int isempty()

{

if(f==r)

return 1;

else

return 0;

}

int isfull()

{

if(r==9)

return 1;

else

return 0;

}

void enqueue()

{

if(isfull())

printf("Queue Overflow\n");

else

{

printf("Enter the element: ");

scanf("%f",&ar[r]);

r=r+1;

printf("\n");

}

}

void dequeue()

{

if(isempty())

printf("Queue Underflow\n");

else

{

printf("%f",ar[f]);

f=f+1;

printf("\n");

}

}

void display()

{

int i;

for(i=f;i<r;i++)

printf("%f \n",ar[i]);

}

void main()

{

int c,ch=1,a=0;

printf("1. isEmpty\n2. isFull\n3. Enqueue\n4. Dequeue\n5. Display\nDefault: exit");

while(ch==1)

{

printf("\nEnter your choice: ");

scanf("%d",&c);

switch (c)

{

case 1:

a=isempty();

if(a==1)

printf("Queue Underflow\n");

else

printf("Queue is not empty\n");

break;

case 2:

a=isfull();

if(a==1)

printf("Queue Overflow\n");

else

printf("Queue is not full\n");

break;

case 3:

enqueue();

break;

case 4:

dequeue();

break;

case 5:

display();

break;

default:

ch = 0;

}

}

}

***Output 1:***

1. isEmpty

2. isFull

3. Enqueue

4. Dequeue

5. Display

Default: exit

Enter your choice: 1

Queue Underflow

Enter your choice: 2

Queue is not full

Enter your choice: 3

Enter the element: 4.7

Enter your choice: 3

Enter the element: 8.9

Enter your choice: 3

Enter the element: 6.7

Enter your choice: 4

4.700000

Enter your choice: 4

8.900000

Enter your choice: 3

Enter the element: 2.567

Enter your choice: 3

Enter the element: 1.9856

Enter your choice: 3

Enter the element: 2.9898

Enter your choice: 3

Enter the element: 12.1

Enter your choice: 3

Enter the element: 23.4

Enter your choice: 3

Enter the element: 14.1

Enter your choice: 3

Enter the element: 2.32

Enter your choice: 3

Queue Overflow

Enter your choice: 5

6.700000

2.567000

1.985600

2.989800

12.100000

23.400000

14.100000

2.320000

Enter your choice: 2

Queue Overflow

Enter your choice:

***Code 2: Infix to Postfix***

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

// stdlib is imported for exit() functionality

// ctype is imported for comparisons like, isdigit()

// string is imported for concatenation of strings

// Basic Operations first.

// Assume maximum length to be 100

char st[100];

int t = -1;

int isempty()

{

if(t==-1)

return 1;

else

return 0;

}

int isfull()

{

if(t==99)

return 1;

else

return 0;

}

void push(char element)

{

if(isfull())

printf("Stack Overflow\n");

else

t++;

st[t]=element;

}

char pop()

{

if(isempty())

printf("Stack Underflow\n");

else

t--;

return st[t+1];

}

// Additional functions

int isop(char s)

{

if(s=='^'|| s=='/'||s=='\*'|| s=='+'||s=='-')

return 1;

else

return 0;

}

int pre(char s)

{

if(s=='^')

return(3);

else if(s=='/' || s=='\*')

return(2);

else if(s=='+' || s=='-')

return(1);

else

return(0);

}

void change(char i[], char p[])

{

int a=0;

int b=0;;

char el;

char c;

push('(');

strcat(i,")");

el=i[a];

while(el!='\0')

{

if(el=='(')

push(el);

else if(isdigit(el) || isalpha(el))

{

p[b]=el;

b++;

}

else if(isop(el)==1)

{

c=pop();

while(isop(c)==1 && pre(c)>=pre(el))

{

p[b]=c;

b++;

c = pop();

}

push(c);

push(el);

}

else if(el==')')

{

c = pop();

while(c!='(')

{

p[b]=c;

b++;

c=pop();

}

}

else

{

printf("\nInvalid Expression.\n");

getchar();

exit(1);

}

a++;

el = i[a];

}

if(t>0)

{

printf("\nInvalid Expression.\n");

getchar();

exit(1);

}

p[b]='\0';

}

int main()

{

char infix[100],postfix[100];

printf("\nEnter Infix: ");

gets(infix);

change(infix,postfix);

printf("Postfix Expression: ");

puts(postfix);

// Normal scanf and printf didn't work

// Hence gets and puts keywords had to be used

return 0;

}

***Output 2:***

Enter Infix: A+(B\*C-(D/E^F)\*G)\*H

Postfix Expression: ABC\*DEF^/G\*-H\*+