Q1

#include<stdio.h>  
#include<stdlib.h>  
#define size 5  
struct stack  
{  
int s[size];  
int top;  
}st;  
  
void push()  
{int item;  
  
scanf("%d",&item);  
st.top++;  
st.s[st.top]=item;}  
void pop()  
{  
int item=st.s[st.top];  
st.top--;}  
int main()  
{int n,c;  
printf("to continue press 1");  
scanf("%d",&n);  
while(n==1)  
{  
printf("to push press 1 and to pop press 2");  
scanf("%d",&c);  
if(c==1){if(st.top>size-1)  
return 0;  
push();}  
else if(c==2){if(st.top<0)  
return 0;  
pop();}  
else   
return 0;  
printf("to continue press 1");  
scanf("%d",&n);  
}  
for(int i=st.top;i>0;i--)  
printf("%d",st.s[i]);  
}

Q2

#include<stdio.h>

#include<stdlib.h>

#include<malloc.h>

struct node

{

int info;

struct node \*ptr;

}\*top=NULL;

void push(int);

void pop();

void display();

void main()

{

int choice,value;

while(1){

printf("enter your choice");

scanf("%d",&choice);

switch(choice){

case 1: printf("enter the value to be insert:");

scanf("%d",&value);

push(value);

break;

case 2: pop(); break;

case 3: display(); break;

case 4: exit(0);

}

}

}

void push(int value)

{

struct node \*newnode;

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

newnode->info=value;

if(top == NULL)

newnode->ptr=NULL;

else

newnode->ptr=top;

top=newnode;

}

void pop()

{

if(top==NULL)

printf("stack is empty");

else

{

struct node \*temp=top;

printf("deleted element: %d",temp->info);

top=temp->ptr;

free(temp);

}

}

void display()

{

if(top==NULL)

printf("stack is empty");

else

{

struct node \*temp=top;

while(temp->ptr !=NULL){

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

temp=temp->ptr;

}

printf("%d NULL",temp->info);

}

}

Q3

#include<stdio.h>

#include<stdlib.h>

#define size 5

struct stack

{

int s[size];

int top;

}ste,sto;

void push()

{int item;

printf("enter the item");

scanf("%d",&item);

if(item%2==0){

ste.top++;

ste.s[ste.top]=item;}

else

{

sto.top++;

sto.s[sto.top]=item;}}

void pop()

{while(ste.top>0){

int item=ste.s[ste.top];

printf("%d ",item);

ste.top--;}

while(sto.top>0){

int item=sto.s[sto.top];

printf("%d ",item);

sto.top--;}}

int main()

{int n,c;

printf("to continue press 1");

scanf("%d",&n);

while(n==1)

{

if((ste.top>size-1)||(sto.top>size-1))

return 0;

push();

printf("to continue press 1");

scanf("%d",&n);}

if((ste.top<0)||(ste.top<0))

return 0;

pop();}

6

#include<stdio.h>

char stack[20];

int top = -1;

void push(char x)

{

    stack[++top] = x;

}

char pop()

{

    if(top == -1)

        return -1;

    else

        return stack[top--];

}

int priority(char x)

{

    if(x == '(')

        return 0;

    if(x == '+' || x == '-')

        return 1;

    if(x == '\*' || x == '/')

        return 2;

}

main()

{

    char exp[20];

    char \*e, x;

    printf("Enter the expression :: ");

    scanf("%s",exp);

    e = exp;

    while(\*e != '\0')

    {

        if(isalnum(\*e))

            printf("%c",\*e);

        else if(\*e == '(')

            push(\*e);

        else if(\*e == ')')

        {

            while((x = pop()) != '(')

                printf("%c", x);

        }

        else

        {

            while(priority(stack[top]) >= priority(\*e))

                printf("%c",pop());

            push(\*e);

        }

        e++;

    }

    while(top != -1)

    {

        printf("%c",pop());

    }

}