Stack Lab using C++

```
#include<iostream>
using namespace std;
#define N 10
class Stack{
    private:
        int data[20];
        int top=-1;
    public:
        void push(int d){
          if(top==N-1){
              cout<<"Stack overflow"<<endl;
          }
          else{
              ++top;
              data[top]=d;
              cout<<"Element: " << data[top] <<"
inserted."<<endl;
          }
        }
        void pop(){
           if(top==-1){
               cout<<"Stack underflow"<<endl;
           }
           else{
              int d;
              d=data[top];
              top--;
              cout<<"The popped element is "<<d<<endl;
           }
        }
        void display(){
           if(top==-1){
              cout<<"Stack empty"<<endl;
           }
           else{

           }
           for(int i=top;i>=0;i--)
           {
              cout<<data[i]<<"\t";
           }
           cout<<endl;
        }

};
```

```
int main()
{
    Stack s;
    int choice,d;
    while(1){
    cout<<"1. Push elements in the stack:"<<endl;
    cout<<"2. Pop elements from the stack:"<<endl;
    cout<<"3. Display elements in the stack:"<<endl;
    cout<<"Enter choice"<<endl;
    cin>>choice;
    switch(choice){
        case 1: {
            cout<<"Enter elements to be added:";
            cin>>d;
            s.push(d);
            break;
        }
        case 2: {
            s.pop();
            break;
    }
        case 3: {
            s.display();
            break;
        }
        default:{
            exit(0);
        }
    }
    }
    return 0;
}
```

```
//Infix to postfix

#include <iostream>

using namespace std;
//#include<ctype.h>

char stack[20];
int top = -1;

void push(char x)
{
        stack[++top]=x;
}

char pop()
{
        if(top == -1)
            cout<<"Stack is empty";
        else
                return stack[top--];
}

int priority(char x)
{
        if(x=='(')
                return 0;
        if(x=='+' || x=='-')
                return 1;
        if(x=='*' || x=='/')
                return 2;
}
int main()
{
        char exp[20];
        char *e,x;
        cout<<"Enter the expression:\n";
        cin>>exp;
        e=exp;
        while(*e != '\0')
        {
                if(isalnum(*e))
                        cout<<*e;
                else if(*e=='(')
                        push(*e);
```

Remaining(....)

```
else if(*e == ')')
                {
                        while((x=pop())!='(')
                                cout<<x;
                }
                else
                {
                        while(priority(stack[top])>=priority(*e))
                           cout<<pop();
                        push(*e);
                }
                e++;
        }
        while(top!=-1)
        {
                cout<<pop();
        }
}
```