Task 1 Fractional Numbers (75 marks)

You have to implement a calculator which supports the calculation of fractional numbers. Each operation of the calculator consists of operand(s), result, number of operands, and the symbol of operation performed.

You have to implement a fraction class:

```
class fraction                    (40 Marks)
{
private:
  int *numerator;
  int * denominator;
public:
  ……..
};
```

For fraction class, you need to implement the following:

- Default constructor
- Parameterized constructor
- Copy constructor
- Destructor
- Overload the following operators (+, –, *, /, >>, <<, ==, =, ++, – –)
    - + to add two fractional numbers
    - – to subtract two fractional numbers
    - * to multiply two fractional numbers
    - / to divide two fractional numbers
    - >> (input operator) to take input into fractional number from user
    - << (output operator) to output a fractional number
    - == (comparison operator) to compare two fractional number and will return true if both numbers are equal and false otherwise
    - = (assignment operator)
    - ++ (pre increment and post increment operator) it will return fractional number +1 as result e.g., incrementing 13/8 will result in 21/8
    - -- (pre decrement and post decrement operator) it will return fractional number -1 as result e.g ., decrementing 13/8 will result in 5/8

Then there is an operation class:

```
class operation                    (20 Marks)
{
  private:
     fraction *operands;
     fraction result;
     int operandCount;
     string symbol;
  public:
     …….
};
```

For operation class, you have to write:

- Default constructor
- Parameterized constructor(s)
- Copy constructor
- Destructor
- Overload following operators (>>, <<)
    - << (output operator) to output operand(s), symbol of operation performed and result.
    - >> (input operator) to take input into operands of a particular operation

A user can:

- Add two fractional numbers using + operator
- Subtract two fractional numbers using – operator
- Multiply two fractional numbers using * operator
- Divide two fractional numbers using / operator
- Compare two fractional numbers using == operator
- Increment/decrement a fractional number using ++/-- operator

You have to use '<<' and '>>' operators to output and input fractional numbers respectively.

In the end you will have a class stack which contains list of performed operations in reverse chronological order.

```
class stack                          (15 Marks)
{
    private:
       operation *list;
       int size;
    public:
          …….
};
```

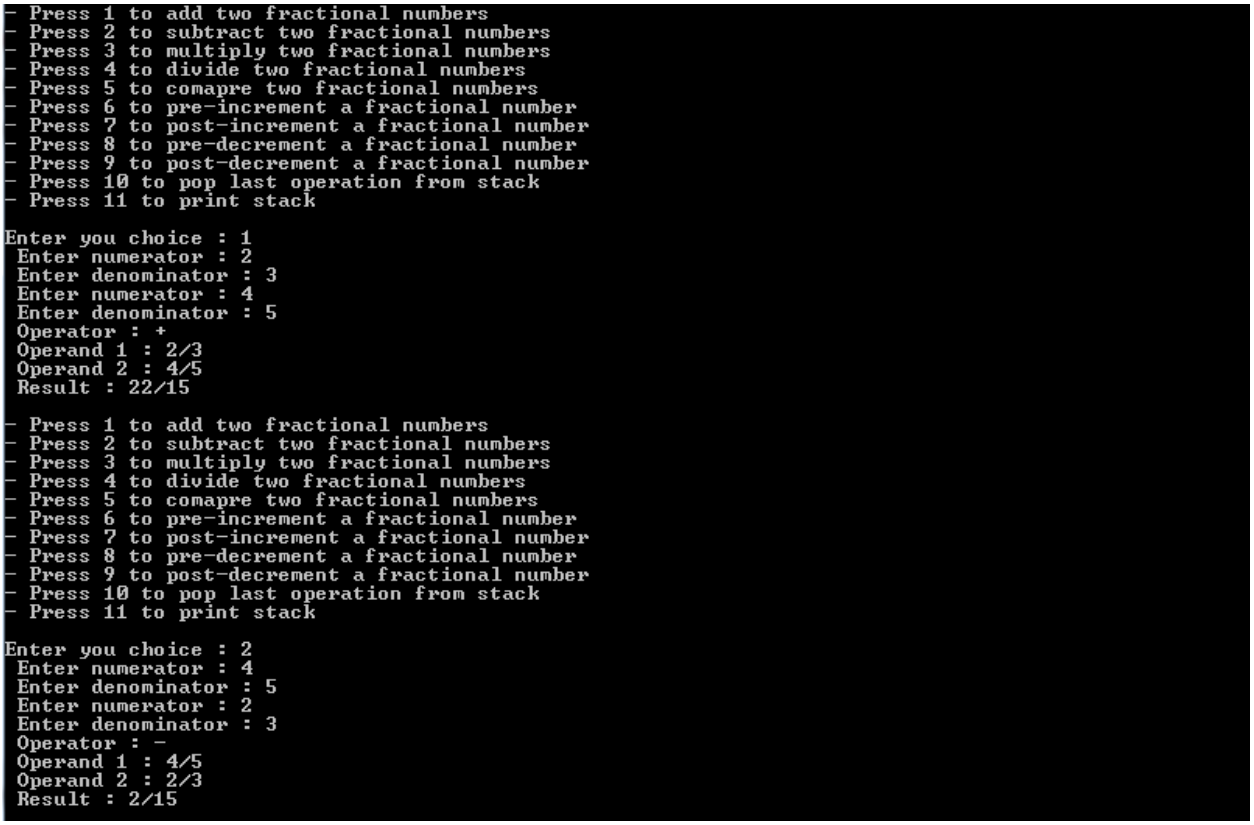
For stack class, you have to write:

- Default constructor
- Parameterized constructor (if necessary)
- Copy constructor (if necessary)
- Destructor
- Push function // used to insert an operation into list of operations
- Pop function // used to remove last inserted operation from list of operations
- Print function // used to print all list of operations in chronological order

Each performed operation will be pushed onto the stack (i.e., added to the list of operations) through the function push (….) and the function pop () that will remove the last inserted operation from the stack.

Make sure that you are not doing 'shallow copy' at any point during this task and in stack class, the list of operations must be resized whenever a new operation is added or an operation is removed through the pop function.

Sample Run:


- Press 1 to add two fractional numbers
- Press 2 to subtract two fractional numbers
- Press 3 to multiply two fractional numbers
- Press 4 to divide two fractional numbers
- Press 5 to comapre two fractional numbers
- Press 6 to pre-increment a fractional number
- Press 7 to post-increment a fractional number
- Press 8 to pre-decrement a fractional number
- Press 9 to post-decrement a fractional number
- Press 10 to pop last operation from stack
- Press 11 to print stack
Enter you choice : 1
Enter numerator : 2
Enter denominator : 3
Enter numerator : 4
Enter denominator : 5
Operator : +
Operand 1 : 2/3
Operand 2 : 4/5
Result : 22/15
- Press 1 to add two fractional numbers
- Press 2 to subtract two fractional numbers
- Press 3 to multiply two fractional numbers
- Press 4 to divide two fractional numbers
- Press 5 to comapre two fractional numbers
- Press 6 to pre-increment a fractional number
- Press 7 to post-increment a fractional number
- Press 8 to pre-decrement a fractional number
- Press 9 to post-decrement a fractional number
- Press 10 to pop last operation from stack
- Press 11 to print stack
Enter you choice : 2
Enter numerator : 4
Enter denominator : 5
Enter numerator : 2
Enter denominator : 3
Operator : -
Operand 1 : 4/5
Operand 2 : 2/3
Result : 2/15

```
- Press 1 to add two fractional numbers
- Press 2 to subtract two fractional numbers
- Press 3 to multiply two fractional numbers
- Press 4 to divide two fractional numbers
- Press 5 to comapre two fractional numbers
- Press 6 to pre-increment a fractional number
- Press 7 to post-increment a fractional number
- Press 8 to pre-decrement a fractional number
- Press 9 to post-decrement a fractional number
- Press 10 to pop last operation from stack
- Press 11 to print stack

Enter you choice : 3
 Enter numerator : 1
 Enter denominator : 3
 Enter numerator : 11
 Enter denominator : 2
 Operator : *
 Operand 1 : 1/3
 Operand 2 : 11/2
 Result : 11/6

- Press 1 to add two fractional numbers
- Press 2 to subtract two fractional numbers
- Press 3 to multiply two fractional numbers
- Press 4 to divide two fractional numbers
- Press 5 to comapre two fractional numbers
- Press 6 to pre-increment a fractional number
- Press 7 to post-increment a fractional number
- Press 8 to pre-decrement a fractional number
- Press 9 to post-decrement a fractional number
- Press 10 to pop last operation from stack
- Press 11 to print stack

Enter you choice : 11
 *** List of Performed Operations ***
 Operator : *
 Operand 1 : 1/3
 Operand 2 : 11/2
 Result : 11/6

 Operator : -
 Operand 1 : 4/5
 Operand 2 : 2/3
 Result : 2/15

 Operator : +
 Operand 1 : 2/3
 Operand 2 : 4/5
 Result : 22/15
```