```cpp
/*
    Programma : cppVerificaClasseFrazione.cpp
    Autore : Riontino Raffaele 4 informatici corso serale
            ITIS Ettore Molinari - Milano 21/12/2010
            Traccia : Scrivere un main che utilizzi la suddetta classe frazione e che
            svolga le seguenti operazioni:
            1. Crea un array di Frazioni (punti 1)
            2. Carichi le frazioni (punti 1)
            3. Visualizzi le frazioni caricate nel vettore (punti 2)
            4. Cerchi e Visualizzi il MAX ed il MIN tra le frazioni
            caricate nell'array (punti 2)
            Aggiungere l'implementazione degli operatori: <, >=, ==, !=.

    aggiornamento : corrette alcune visualizzazioni di troppo.
*/
#include <iostream>
#include <conio.h>
using namespace std;
class frazione
    {
    public : //metodi e proprietà publiche dell'oggetto frazione
            frazione (); //costruttore della frazione
            frazione (int); //costruttore con un parametro di default da assegnare al numeratore
            frazione (int , int); //costruttore con due parametri di default
            void input (); //metodo per l'inserimento dei valori della frazione
            void stampa () {cout << this -> num << "/" << this -> den;} //metodo per la visualizzazione della frazione
            frazione operator + (frazione); // implementazione dell'operatore +
            void operator += (frazione temp) {*this = *this + temp;} //implementazione dell'operatore +=
            frazione operator - (frazione); // implementazione dell'operatore -
            void operator -= (frazione temp) {*this = *this - temp;} //implementazione dell'operatore -=
            frazione operator * (frazione); // implementazione dell'operatore *
            void operator *= (frazione temp) {*this = *this * temp;} //implementazione dell'operatore *=
            frazione operator / (frazione); //implementazione dell'operatore /
            void operator /= (frazione temp) {*this = *this / temp;} // implementazione dell'operatore /=
            void operator = (frazione); // implementazione dell'operatore =
            bool operator > (frazione); // implementazione dell'operatore >
            bool operator < (frazione); // implementazione dell'operatore <
            bool operator >= (frazione); // implementazione dell'operatore >=
            bool operator <= (frazione); // implementazione dell'operatore <=
            bool operator == (frazione); // implementazione dell'operatore ==
            bool operator != (frazione); // implementazione dell'operatore !=
            ~frazione (); //distruttore
    private : //metodi e proprietà utilizzabili solo dalla classe
            int num ; //numeratore della frazione
            int den ; //denominatore della frazione
            string controlla (string); //metodo che controlla se sono stati inseriti solo numeri o il segno meno
            void segno (); // metodo che gestisce il segno negativo nel caso si trovi al denominatore
            void riduci (); //metodo che semplifica la frazione ai minimi termini
    }; //fine header class

////////////////COSTRUTTORI PUBBLICI//////////////////////////
frazione::frazione() //costruttore della frazione senza parametri
    {
    this -> num = 0; //assegno il numeratore di default uguale a 0
    this -> den = 1; //assegno il denominatore di default uguale a 1
    }

frazione::frazione(int temp)
    {
    this -> num = temp;
    this -> den = 1;
    (*this).segno();
    (*this).riduci();
    }

frazione::frazione(int n , int d)
    {
    this -> num = n;
    if (d == 0)
        {
        string temp;
        bool ripeti = true;
        cout << "Attenzione, e' stato inserito il valore 0 al denominatore!!\n";
        do{
            cout << "Reinserisci il denominatore : ";
            cin >> temp;
            temp = controlla(temp);
          }while(temp == "errore" || atoi(temp.c_str()) == 0);
        d = atoi(temp.c_str());
        }
    this -> den = d;
    (*this).segno();
    (*this).riduci();
    }

frazione::~frazione() //distruttore
    {
    }
```

```
void frazione::input ()
    {
    string temp;
    do {
        cout << "inserisci il numeratore : ";
        cin >> temp;
        temp = controlla(temp);
        if (temp == "errore") cout << "re";
       }while (temp == "errore");
    this -> num = atoi(temp.c_str());
    do {
        cout << "inserisci il denominatore : ";
        cin >> temp;
        temp = controlla(temp);
        if (atoi(temp.c_str()) == 0) cout << "Valore non consentito!!\nre";
       }while (temp == "errore" || atoi(temp.c_str()) == 0);
    this -> den = atoi(temp.c_str());
    (*this).segno();
    (*this).riduci();
    }

frazione frazione::operator + (frazione temp)
    {
    frazione ris;
    ris.num = (this -> num * temp.den) + (this -> den * temp.num);
    ris.den = this -> den * temp.den;
    ris.riduci();
    return ris;
    }

frazione frazione::operator - (frazione temp)
    {
    frazione ris;
    ris.num = (this -> num * temp.den) - (this -> den * temp.num);
    ris.den = this -> den * temp.den;
    ris.riduci();
    return ris;
    }

frazione frazione::operator * (frazione temp)
    {
    frazione ris;
    ris.num = this -> num * temp.num;
    ris.den = this -> den * temp.den;
    ris.riduci();
    return ris;
    }

frazione frazione::operator / (frazione temp)
    {
    frazione ris;
    ris.num = this -> num * temp.den;
    ris.den = this -> den * temp.num;
    ris.riduci();
    return ris;
    }

bool frazione::operator > (frazione temp)
    {
    if ((this -> num * temp.den) > (this -> den * temp.num))
    return true;
    else return false;
    }

bool frazione::operator < (frazione temp)
    {
    if ((this -> num * temp.den) < (this -> den * temp.num))
    return true;
    else return false;
    }

bool frazione::operator >= (frazione temp)
    {
    if ((this -> num * temp.den) >= (this -> den * temp.num))
    return true;
    else return false;
    }

bool frazione::operator <= (frazione temp)
    {
    if ((this -> num * temp.den) <= (this -> den * temp.num))
    return true;
    else return false;
    }

bool frazione::operator == (frazione temp)
    {
```

```
    if ((this -> num * temp.den) == (this -> den * temp.num))
    return true;
    else return false;
    }

bool frazione::operator != (frazione temp)
    {
    if ((this -> num * temp.den) != (this -> den * temp.num))
    return true;
    else return false;
    }

void frazione::operator = (frazione temp)
    {
    this -> num = temp.num;
    this -> den = temp.den;
    }
////////////////////////////////FINE PUBBLICI//////////////////////////////
//////////////////////COSTRUTTORI PRIVATI//////////////////////

string frazione::controlla(string temp)
    {
    for (int i = 0 ; i < temp.length() ; i++)
        {
        if (!isdigit(temp.at(i)) && temp.at(i) != '-')
            {
            cout << "Caratteri non consentiti!!\n";
            return "errore";
            }
        }
    return temp;
    }

void frazione::segno()
    {
    if (this -> den < 0)
        {
        this -> num *= -1;
        this -> den *= -1;
        }
    }

void frazione::riduci()
    {
    int min , segno;
    if (this -> num < 0)
        {
        segno = -1;
        this -> num *= -1;
        }
    else segno = 1;
    bool continua = true;
    do{
        continua = false;
        if (this -> num <= this -> den) min = this -> num;
        else min = this -> den;
        for (int i = 2 ; i <= min ; i ++)
            {
            if (this -> num % i == 0 && this -> den % i == 0)
                {
                this -> num /= i;
                this -> den /= i;
                continua = true;
                }
            }
      }while (continua);
    this -> num *=segno;
    }
////////////////////////FINE PRIVATI////////////////////////////////

void titolo();

char menu();

void visualizza(frazione*,int);

int readInt(string);

void carica(frazione*&,int);

frazione maggiore(frazione*,int);

frazione minore(frazione*,int);


int main ()
{
system("color F1");
```

```
char scelta;
int dim = 0;
frazione *dati = NULL;
frazione max,min;
do{
    scelta = menu();
    switch(scelta)
        {
        case 'v' :  if (dim == 0)
                        cout << "\n\n\tL'array non e' stato caricato!!\n";
                    else
                        visualizza(dati,dim);
                    getch();
                    break;
        case 'c' :  char s ;
                    s = 's';
                    if (dim != 0)
                        {
                        cout << "\n\tL'array contiene dei valori, eliminarli (s / n)? \n";
                        s = getch();
                        }
                    if (s == 's')
                        {
                        do{
                            dim = readInt("\n\tInserisci la dimansione dell'array : ");
                            if (dim < 2) cout << "\tAttenzione!! La dimensione deve essere maggiore di 1!!\n";;
                          }while(dim < 2);
                        delete dati;
                        dati = new frazione[dim];
                        carica(dati,dim);
                        }
                    break;
        case 'g' :  if (dim == 0)
                        cout << "\n\n\tL'array di frazioni non e' stato caricato!!";
                    else {
                            titolo();
                            max = maggiore(dati,dim);
                            cout << "\n\n";
                            visualizza(dati,dim);
                            cout << "\n\n\tla frazione piu' grande e' : " ; max.stampa();
                         }
                    getch();
                    break;
        case 'p' :  if (dim == 0)
                        cout << "\n\n\tL'array di frazioni non e' stato caricato!!";
                    else {
                            titolo();
                            min = minore(dati,dim);
                            cout << "\n\n";
                            visualizza(dati,dim);
                            cout << "\n\n\tla frazione piu' piccola e' : " ; min.stampa();
                         }
                    getch();
                    break;
        }
   }while (scelta != 'e');
return 1;
}//fine main


void titolo()
    {
    system("cls");
    cout << "\n\t\tVerifica sulla classe frazione\t21/12/2010" ;
    cout << "\n\n Riontino Raffaele classe 4 inf. ITIS Molinari - Milano (corso serale)\n\n";
    }//fine titolo


char menu()
   {
    char s;
    titolo();
    cout << "\n\t\t\tMENU'\n";
    cout << "\n\t[c] : carica / ricarica l'array di frazioni\n";
    cout << "\n\t[v] : visualizza l'array di frazioni\n";
    cout << "\n\t[g] : visualizza la frazione maggiore inserita\n";
    cout << "\n\t[p] : visualizza la frazione minore inserita\n";
    cout << "\n\t[e] : esci.\n";
    s = getch();
    s = tolower(s);
    return s;
   } //fine menu


void visualizza(frazione* dati,int dim)
    {
    titolo();
    cout << "\n\n\tFrazioni caricate : \n\n";
```

```
    if (dim == 0) {
                    cout << "\n\n\tL'array e' vuoto!!";
                    return;
                  }
    for (int i = 0 ; i < dim ; i ++)
        {
        cout << "\t";
        (*(dati+i)).stampa();
        }
    } //fine visulaizza


int readInt(string messaggio)
    {
    bool continua;
    string temp;
    int i ;
    do{
        cout << messaggio;
        cin >> temp;
        i = 0;
        continua = true;
        while(i < temp.length() && continua)
            {
            if (!isdigit(temp.at(i)) && temp.at(i) != '-')
                {
                cout << "Caratteri non consentiti!!\n";
                temp = "errore";
                continua = false;
                }
            i++;
            }
      }while(temp == "errore");
    return (atoi(temp.c_str()));
    } //fine readInt


void carica(frazione*& dati,int dim)
    {
    for (int i = 0 ; i < dim ; i++)
        {
        titolo();
        if (i > 0) visualizza (dati,i);
        cout << "\n\n\tcaricamento dei dati \n\n";
        cout << "\nfrazione n " << (i+1) << " : \n";
        dati[i].input();
        }
    titolo();
    visualizza(dati,dim);
    cout << "\n\n\tCaricamento terminato!!\n";
    getch();
    }//fine carica


frazione maggiore(frazione* dati,int dim)
    {
    frazione max = dati[0];
    for (int i = 0 ; i < dim ; i ++)
        if ((*(dati+i)) > max) max = (*(dati+i));
    return max;
    }//fine maggiore


frazione minore(frazione* dati,int dim)
    {
    frazione min = dati[0];
    for (int i = 0 ; i < dim ; i ++)
        if ((*(dati+i)) < min) min = (*(dati+i));
    return min;
    }//fine minore
```