

Esercitazione n. 5

Modificare la classe `CMyImage` sviluppata nell'Esercitazione 2 aggiungendo opportuni operatori sovraccaricati per gestire operazioni su immagini, ad esempio:

- Differenza tra immagini (operatore -)
- Soglia di immagini (operatore <)
- Stampa a video delle informazioni di un'immagine (tipo e dimensioni) (operatore <<)
- Accesso ai singoli pixels dell'immagine secondo la sintassi `im(i, j)` (pixel presente alla riga *i*-esima e colonna *j*-esima)
- Moltiplicazione di tutti i pixels di un'immagine per uno scalare.

Verificare la correttezza dell'implementazione mediante i seguenti test e utilizzare la classe `CMyViewer` sviluppata nell'esercitazione 4 per visualizzare le immagini.

```
void test1()
{
    cout<<"Test overloading operator <<"<<endl;
    CMyImage a("im1.pgm");
    cout<<a;
}
```

```
void test2()
{
    cout<<"Test overloading operator -"<<endl;
    CMyImage a("im1.pgm");
    CMyImage b("im2.pgm");
    CMyImage res= a-b;
    res.WriteData("diff.pgm");
}
```

```

void test3()
{
    cout<<"Test overloading operator ()"<<endl;
    CMyImage a("im1.pgm");
    for(int i=0; i<100; i++)
        for(int j=20;j<100;j++)
            a(i,j)=0;
    a.WriteData("black.pgm");
}

void test4()
{
    cout<<"Test overloading operator <"<<endl;
    CMyImage a("im1.pgm");
    CMyImage res=a<50;
    res.WriteData("threshold.pgm");
}

void test5()
{
    cout<<"Test overloading operator *"<<endl;
    CMyImage a("im1.pgm");
    int minout=200;
    int maxout=250;
    CMyImage res=(a-0)*((maxout-minout)/(255.0))+minout;
    res.WriteData("stretch.pgm");
}

void test6()
{
    cout<<"Test copy constructor"<<endl;
    CMyImage a("im1.pgm");
    CMyImage b=a;
    a.WriteData("a1.pgm");
    b.WriteData("b1.pgm");
    for(int i=0; i<100; i++)
        for(int j=20;j<100;j++)
            a(i,j)=0;
    a.WriteData("a2.pgm");
    b.WriteData("b2.pgm");
}

```