

```

#include <iostream>
#include <string.h>

using namespace std;

const unsigned DIM_MIN=3;
const unsigned MIN_LIN=2;
const unsigned MIN_COL=2;
class Vector
{
    //declararea clasei Matrice ca prieten a clasei Vector
    friend class Matrice;
protected:
    double *ptrData;
    unsigned dimensiune;
public:
    Vector(unsigned Dimensiune=DIM_MIN)
        {ptrData=new double [dimensiune=Dimensiune];}
    ~Vector()
    {delete [] ptrData;}
    unsigned intoarceDimensiune()
        {return dimensiune;}
    double& operator[] (unsigned indice)
        {return *(ptrData+indice);}
};
class Matrice
{
protected:
    double *ptrData;
    unsigned maxLin;
    unsigned maxCol;
public:
    Matrice(unsigned Linii=MIN_LIN,unsigned Coloane=MIN_COL)
:
    maxLin(Linii), maxCol(Coloane)
    {ptrData=new double [Linii*Coloane];}
    ~Matrice() {delete [] ptrData;}
    unsigned intoarceMaxLinii() {return maxLin;}
    unsigned intoarceMaxColoane() {return maxCol;}
    double& operator() (unsigned lin, unsigned col)
        {return *(ptrData+lin+col*maxLin);}
    void copiazLinie(Vector& tab, unsigned lin);};
    void Matrice::copiazLinie(Vector& tab,unsigned lin)
    {
        //eliberarea spatiului de memorie rezervat tabloului si
        //refacerea lui cu maxCol elemente
        delete [] tab.ptrData; //=maxCol;
        tab.ptrData=new double [tab.dimensiune];
        for(unsigned col=0;col<maxCol;col++)
            tab[col]=*(ptrData+lin+col*maxLin);
    }
};

```

```

    }

int main()
{
    const unsigned DIM_TAB=5;const unsigned LINII=3;
    const unsigned COL=3;
    unsigned lin,col;
    Vector vect(DIM_TAB);
    Matrice mat(LINII,COL);
    //atribuie valori elementelor tabloului vect
    for(unsigned i=0;i<vect.intoarceDimensiune();i++)
vect[i]=2.5+i*i;
    //atribuie valori elementelor matricei
    for(lin=0;lin<mat.intoarceMaxLinii();lin++)
        for(col=0;col<mat.intoarceMaxColoane();col++)
            mat(lin,col)=5.5+lin+10*col;
    cout<<"Vectorul vect are urmatoarele elemente: \n\n";
    for(unsigned i=0;i<vect.intoarceDimensiune();i++)
        cout<<"vect["<<i<<"]="<<vect[i]<<"\n";
    cout<<"Apasati o tasta pentru a continua...";
    //getch();
    cout<<"Matricea mat are urmatoarele elemente: \n\n";
    for(lin=0;lin<mat.intoarceMaxLinii();lin++)
        for(col=0;col<mat.intoarceMaxColoane();col++)

    cout<<"mat["<<lin<<","<<col<<"]="<<mat(lin,col)<<"\n";
    cout<<"Apasati o tasta pentru a continua...";
    //getch();
    //copierea liniei 0 din matricea mat in vectorul vect
    mat.copiazLinie(vect,0);
    cout<<"Vectorul vect are urmatoarele elemente: \n\n";
    for(unsigned i=0;i<vect.intoarceDimensiune();i++)
        cout<<"vect["<<i<<"]="<<vect[i]<<"\n";
    return 0;
}

```