```cpp
// exam2007.C

#include <iostream>
using namespace std;

const int N = 4;

class Vect
{
public :
    Vect ();
    Vect (const int* tab);
    friend ostream& operator<< (ostream& flot, const Vect& vec);
    int get (int i) const       { return elem[i]; }
    void set (int i, int val)   { elem[i] = val; }
    Vect operator+ (const Vect& vec2) const;
    int operator* (const Vect& vec2) const;

private :
    int elem[N];
};

Vect::Vect ()
{
    for (int i = 0; i < N; i++)
        elem[i] = 0;
}

Vect::Vect (const int* tab)
{
    for (int i = 0; i < N; i++)
        elem[i] = tab[i];
}

ostream& operator<< (ostream& flot, const Vect& vec)
{
    cout << "(";

    for (int i = 0; i < N; i++)
        cout << vec.elem[i] << " ";

    cout << ")";

    return flot;
}

Vect Vect::operator+ (const Vect& vec2) const
{
    Vect somme;

    for (int i = 0; i < N; i++)
        somme.elem[i] = elem[i] + vec2.elem[i];

    return somme;
}

int Vect::operator* (const Vect& vec2) const
{
    int produit = 0;

    for (int i = 0; i < N; i++)
        produit += elem[i] * vec2.elem[i];

    return produit;
}

class Tab2D
{
public :
    Tab2D (int val);
    void afficher () const;
    int get (int i, int j) const        { return elem[i][j]; }
```

```
    void set (int i, int j, int val)    { elem[i][j] = val; }

private :
    int elem[N][N];
};

Tab2D::Tab2D (int val)
{
    for (int i = 0; i < N; i++)
        for (int j  = 0; j < N; j++)
            elem[i][j] = val;
}

void Tab2D::afficher () const
{
    cout << endl;

    for (int i = 0; i < N; i++)
    {
        // affichage d'une ligne
        for (int j  = 0; j < N; j++)
            cout << elem[i][j] << " ";
        cout << endl;
    }
}

class Mat : public Tab2D
{
public :
    Mat ();
    Mat (const int tab[N][N]);
    Vect getLigne (int numlig) const;
    Vect getColonne (int numcol) const;
    Mat operator+ (const Mat& mat2) const;
    Mat operator* (const Mat& mat2) const;

private :
};

// dans la liste d'initialisation du constructeur de Mat, appel du constructeur
// de la classe mère Tab2D

Mat::Mat ()
: Tab2D(0)
{
}

Mat::Mat (const int tab[N][N])
    : Tab2D(0)
{
    for (int i = 0; i < N; i++)
        for (int j  = 0; j < N; j++)
            set (i, j, tab[i][j]);
}

Vect Mat::getLigne (int numlig) const
{
    Vect ligne;

    for (int i = 0; i < N; i++)
        ligne.set (i, get(numlig,i));

    return ligne;
}

Vect Mat::getColonne (int numcol) const
{
    Vect colonne;

    for (int i = 0; i < N; i++)
        colonne.set (i, get(i,numcol));
```

```
    return colonne;
}
Mat Mat::operator+ (const Mat& mat2) const
{
    Mat somme;

    for (int i = 0; i < N; i++)
        for (int j  = 0; j < N; j++)
            somme.set(i, j, get(i,j) + mat2.get(i,j));

    return somme;
}

Mat Mat::operator* (const Mat& mat2) const
{
    Mat produit;

    for (int i = 0; i < N; i++)
        for (int j  = 0; j < N; j++)
            produit.set(i, j, getLigne(i) * mat2.getColonne(j));

    return produit;
}

int main ()
{
    /*
    Vect v1;
    cout << v1 << endl;

    int t2[N] = {2, 1, 3, 4};
    Vect v2(t2);
    cout << v2 << endl;

    int t3[N] = {5, 2, 4, 1};
    Vect v3(t3);
    cout << v3 << endl;

    cout << "somme : " << v2 + v3 << endl;
    cout << "produit : " << v2 * v3 << endl;
    */

    Tab2D tab1(2);
    tab1.set (1, 2, 5);
    tab1.afficher();

    Mat m1;
    m1.afficher();

    int t2[N][N] = { {2, 1, 1, 0},
                     {2, 0, 1, 1},
                         {1, 1, 0, 2},
                         {2, 0, 2, 1} };
    Mat m2(t2);
    m2.afficher();

    Vect lig = m2.getLigne (1);
    Vect col = m2.getColonne (1);
    cout << lig << " " << col << endl;

    int t3[N][N] = { {1, 0, 0, 2},
                     {2, 1, 2, 1},
                         {0, 2, 1, 1},
                         {1, 0, 2, 2} };
    Mat m3(t3);
    m3.afficher();

    Mat m4 = m2 + m3;
    m4.afficher();

    Mat m5 = m2 * m3;
    m5.afficher();
```

```
    return 0;
}
```