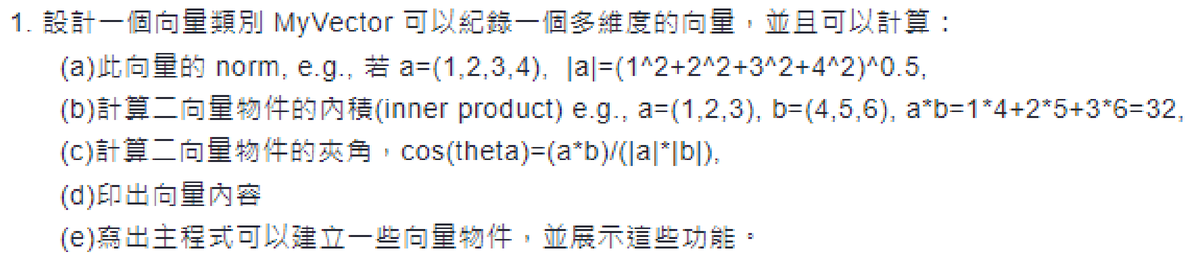
12/01週作業 p1、p2

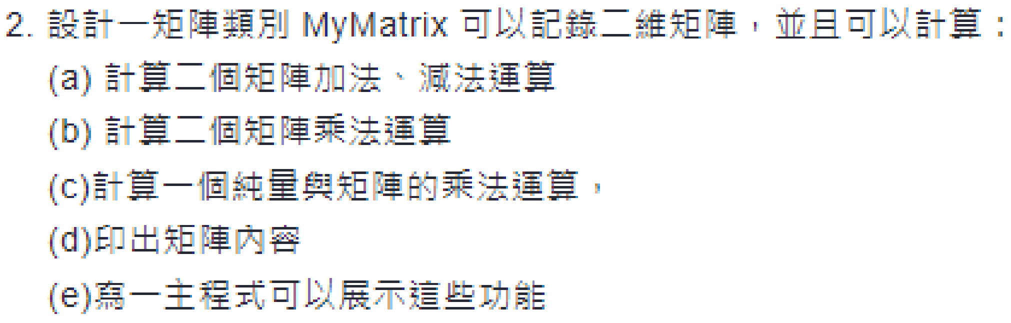


1. 設計一個向量類別 MyVector 可以紀錄一個多維度的向量，並且可以計算：
   (a)此向量的 norm, e.g., 若 a=(1,2,3,4), |a|=(1^2+2^2+3^2+4^2)^0.5,
   (b)計算二向量物件的內積(inner product) e.g., a=(1,2,3), b=(4,5,6), a*b=1*4+2*5+3*6=32,
   (c)計算二向量物件的夾角，cos(theta)=(a*b)/(|a|*|b|),
   (d)印出向量內容
   (e)寫出主程式可以建立一些向量物件，並展示這些功能。

```
/* 12/01  楊育哲
 * 實作第一題: 設計myVector ,可記錄多維度向量
 */
public class h1_1201_w {
    static class MyVector{
        private float[] vector;
        MyVector(float init[]){
            vector = new float[init.length];
            for(int i=0; i<init.length; i++){
                vector[i] = init[i];
            }
        }
        float norm(){
            float ans=0;
            for(int i=0; i<vector.length; i++) ans+=vector[i]*vector[i];
            return ans/2;
        }
        float innorProduct(MyVector b){
            float ans=0;
            if(vector.length!=b.vector.length) return 0;//
            for(int i=0; i<vector.length; i++) ans+=vector[i]*b.vector[i];
            return ans;
        }
        float theta(MyVector b){
            return innorProduct(b)/(norm()*b.norm());
        }
        void Print(){
            System.out.print("(");
            for(int i=0; i<vector.length-1; i++) System.out.printf("%7.2f,", vector[i]);
            System.out.printf("%7.2f)\n", vector[vector.length-1]);
        }
    }
    static public void main(String args[]){
        float[] a = {1, 2, 3};
        MyVector A = new MyVector(a);
        float[] b = {4, 5, 6};
        MyVector B = new MyVector(b);
        A.Print();
        B.Print();
        System.out.printf("A.norm: %f\n", A.norm());
        System.out.printf("B.norm: %f\n", B.norm());
        System.out.printf("innor product : %5.2f\n", A.innorProduct(B));
        System.out.printf("theta : %10.9f\n", A.theta(B));
    }
}
```

輸出:

```
(   1.00,   2.00,   3.00)
(   4.00,   5.00,   6.00)
A.norm: 7.000000
B.norm: 38.500000
innor product : 32.00
theta : 0.118738405
```



2. 設計一矩陣類別 MyMatrix 可以記錄二維矩陣，並且可以計算：

(a) 計算二個矩陣加法、減法運算

(b) 計算二個矩陣乘法運算

(c)計算一個純量與矩陣的乘法運算，

(d)印出矩陣內容

(e)寫一主程式可以展示這些功能

```
/* 1201 楊育哲
 * 實作第二題: myMatrix
 */
public class h2_1201_w {
    static class MyVector{
        private float[] vector;
        MyVector(float init[]){
            vector = new float[init.length];
            for(int i=0; i<init.length; i++){
                vector[i] = init[i];
            }
        }
    }
    static class MyMatrix{
        private float[][] array;
        MyMatrix(float init[][]){
            array = new float[init.length][];
            for(int i=0; i<init.length; i++){
                array[i] = new float[init[i].length];
                for(int j=0; j<init[0].length; j++){
                    array[i][j] = init[i][j];
                }
            }
        }
        MyMatrix Add(MyMatrix b){
            float[][] c = new float[array.length][];
            for(int i=0; i<array.length; i++){
                c[i] = new float[array[i].length];
                for(int j=0; j<array[1].length; j++){
                    c[i][j] = array[i][j]+b.array[i][j];
                }
            }
            MyMatrix C = new MyMatrix(c);
            return C;
        }
        MyMatrix Minus(MyMatrix b){
            float[][] c = new float[array.length][];
            for(int i=0; i<array.length; i++){
                c[i] = new float[array[i].length];
                for(int j=0; j<array[1].length; j++){
                    c[i][j] = array[i][j]-b.array[i][j];
                }
            }
            MyMatrix C = new MyMatrix(c);
            return C;
        }
        MyMatrix Mult_matrix(MyMatrix b){
            int h=Math.max(array.length, array[0].length);
            int w=Math.min(array.length, array[0].length);
            float[][] c = new float[h][];
            for(int i=0; i<h; i++){
                c[i] = new float[h];
                for(int j=0; j<h; j++){
                    for(int k=0; k<w; k++){
                        c[i][j]+=array[i][k]*b.array[k][j];
                    }
                }
            }
            MyMatrix C = new MyMatrix(c);
            return C;
        }
        MyMatrix Mult_vector(MyVector b){
            float[][] c = new float[array.length][];
            for(int i=0; i<array.length; i++){
                c[i] = new float[array[i].length];
                for(int j=0; j<array[1].length; j++){
                    c[i][j] = array[i][j]*b.vector[i];
                }
            }
            MyMatrix C = new MyMatrix(c);
            return C;
        }
        void Print(){
            for(int i=0; i<array.length; i++){
                for(int j=0; j<array[1].length; j++){
                    System.out.printf("%5.2f ", array[i][j]);
                }
                System.out.println("");
            }
        }
    }
    static public void main(String args[]){
        float[][] a = {{1, 2},
                       {3, 4},
                       {5, 6}};
        MyMatrix A = new MyMatrix(a);
        float[][] a_ = {{6, 5},
                       {4, 3},
                       {2, 1}};
        MyMatrix A_ = new MyMatrix(a_);
        float[][] b = {{1, 2, 3},
                       {4, 5, 6}};
        MyMatrix B = new MyMatrix(b);
        float[] c = {2, 3, 4};
        MyVector C = new MyVector(c);
        MyMatrix Ac = A.Mult_vector(C);
        MyMatrix AB = A.Mult_matrix(B);
        MyMatrix F = A_.Add(A);
        MyMatrix G = A_.Minus(A);
        System.out.println("A:");
        A.Print();//原始A矩陣
        System.out.println("A_:");
        A_.Print();//原始A_矩陣
        System.out.println("B:");
        B.Print();//原始B矩陣
        System.out.println("A * C{vector->{2, 3, 4}}:");
        Ac.Print();//A矩陣乘上向量C
        System.out.println("A * B{matrix}:");
        AB.Print();//A矩陣乘上矩陣B
        System.out.println("A_ + A");
        F.Print();//矩陣A_加上矩陣A
        System.out.println("A_ - A");
        G.Print();//矩陣A_減去矩陣A
    }
}
```

輸出:

```
A:
1.00  2.00
3.00  4.00
5.00  6.00
A_:
6.00  5.00
4.00  3.00
2.00  1.00
B:
1.00  2.00  3.00
4.00  5.00  6.00
```

```
A * C{vector->{2, 3, 4}}:
2.00  4.00
9.00 12.00
20.00 24.00
A * B{matrix}:
9.00 12.00 15.00
19.00 26.00 33.00
29.00 40.00 51.00
A_ + A
7.00  7.00
7.00  7.00
7.00  7.00
A_ - A
5.00  3.00
1.00 -1.00
-3.00 -5.00
```