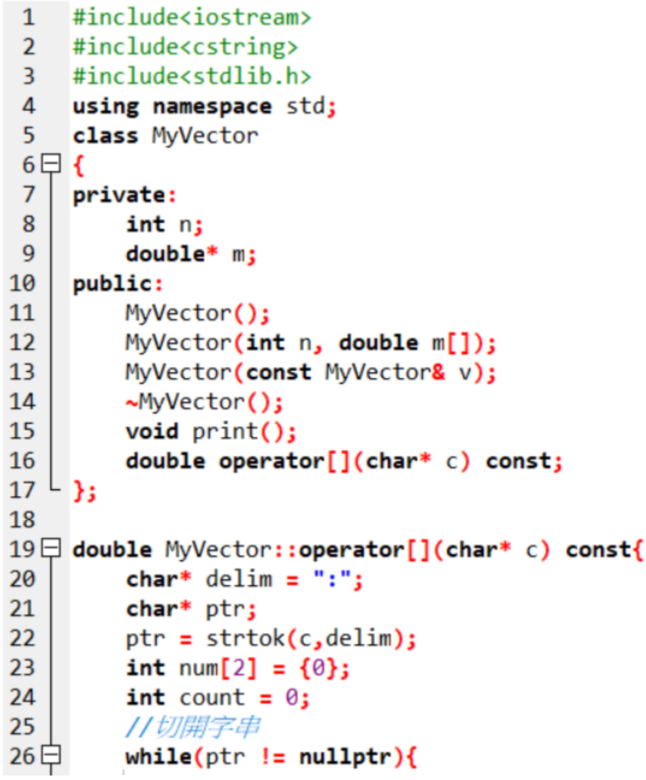
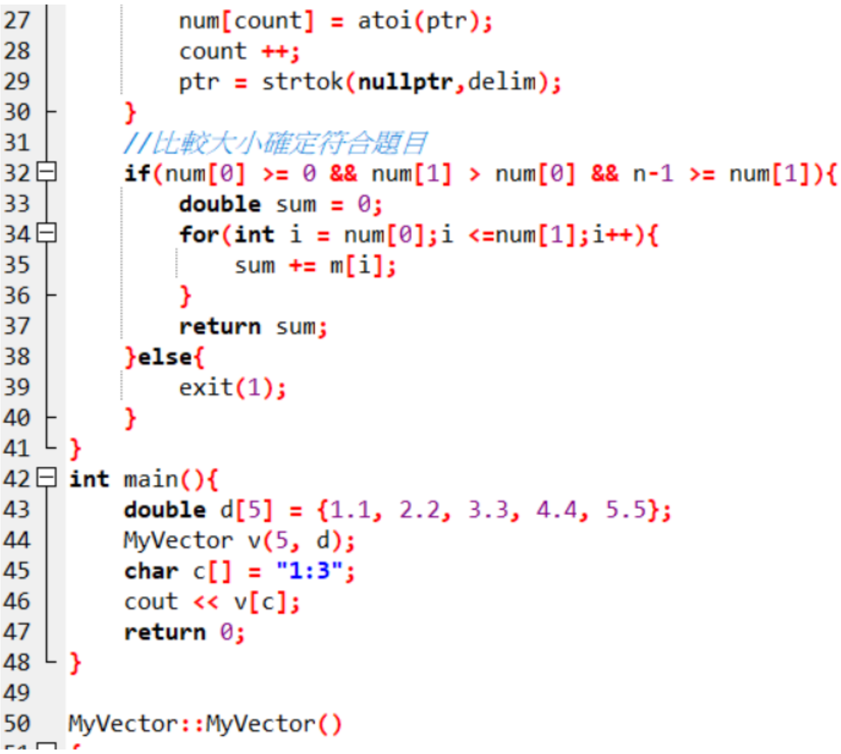
資料結構與進階程式設計作業一

學號:B05605030　系級:森林二　姓名:郭玟妍

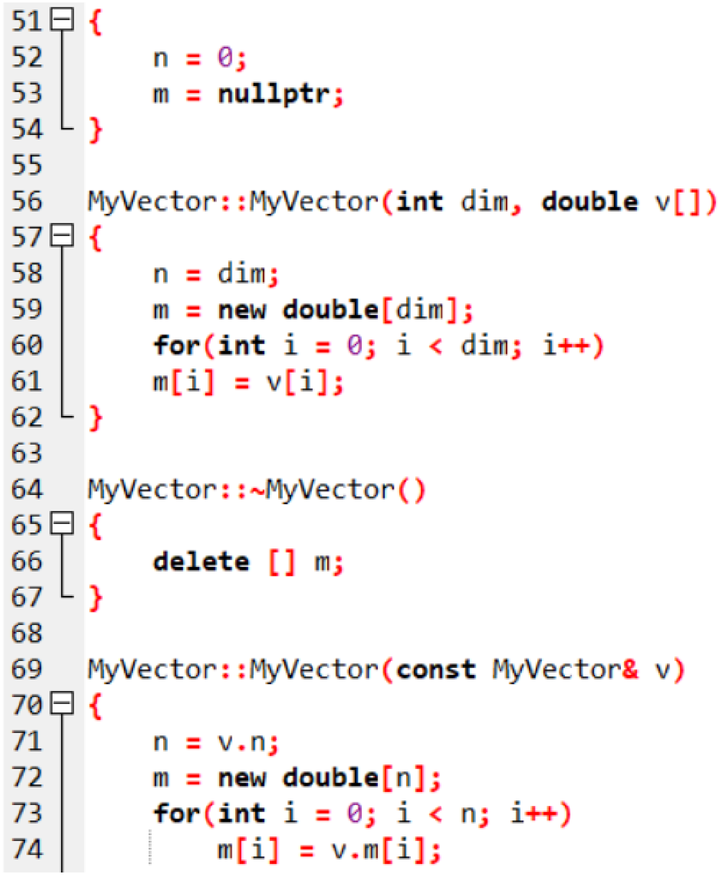
第一題



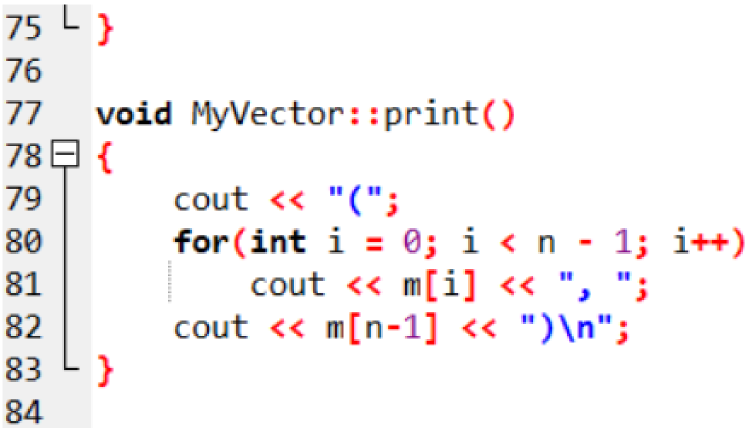
```cpp
#include<iostream>
#include<cstring>
#include<stdlib.h>
using namespace std;
class MyVector
{
private:
    int n;
    double* m;
public:
    MyVector();
    MyVector(int n, double m[]);
    MyVector(const MyVector& v);
    ~MyVector();
    void print();
    double operator[](char* c) const;
};

double MyVector::operator[](char* c) const{
    char* delim = ":";
    char* ptr;
    ptr = strtok(c,delim);
    int num[2] = {0};
    int count = 0;
    //切開字串
    while(ptr != nullptr){
```



```cpp
        num[count] = atoi(ptr);
        count ++;
        ptr = strtok(nullptr,delim);
    }
    //比較大小確定符合題目
    if(num[0] >= 0 && num[1] > num[0] && n-1 >= num[1]){
        double sum = 0;
        for(int i = num[0];i <=num[1];i++){
            sum += m[i];
        }
        return sum;
    }else{
        exit(1);
    }
}
int main(){
    double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
    MyVector v(5, d);
    char c[] = "1:3";
    cout << v[c];
    return 0;
}

MyVector::MyVector()
{
```



```
{
    n = 0;
    m = nullptr;
}

MyVector::MyVector(int dim, double v[])
{
    n = dim;
    m = new double[dim];
    for(int i = 0; i < dim; i++)
    m[i] = v[i];
}

MyVector::~MyVector()
{
    delete [] m;
}

MyVector::MyVector(const MyVector& v)
{
    n = v.n;
    m = new double[n];
    for(int i = 0; i < n; i++)
        m[i] = v.m[i];
```

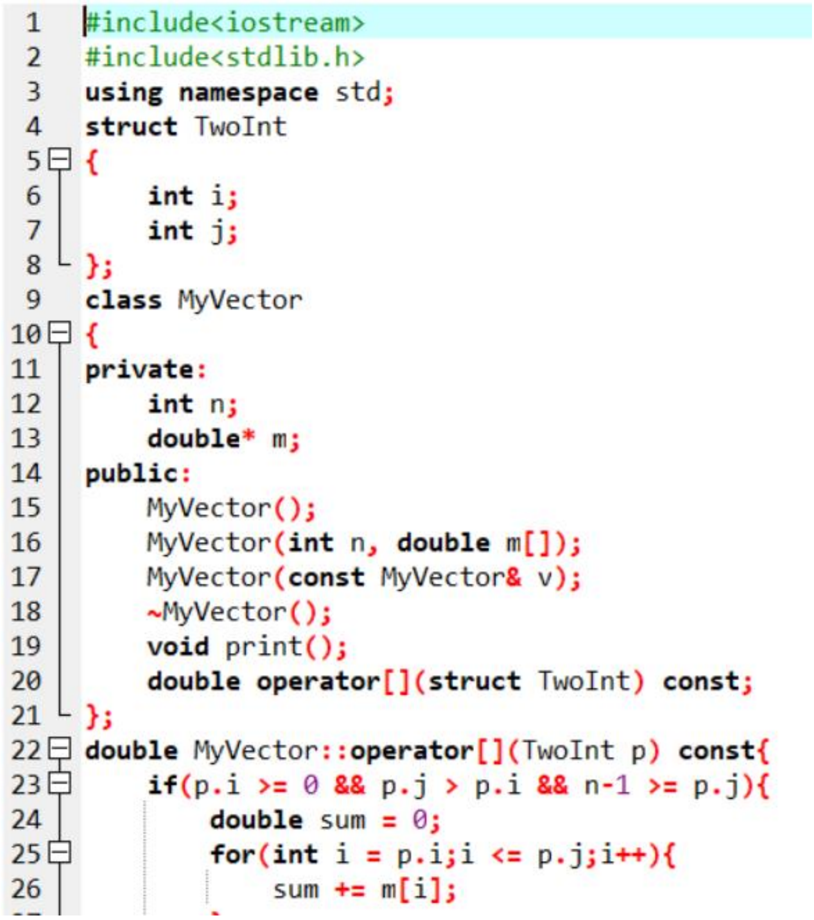


```
}

void MyVector::print()
{
    cout << "(";
    for(int i = 0; i < n - 1; i++)
        cout << m[i] << ", ";
    cout << m[n-1] << ")\n";
}

```

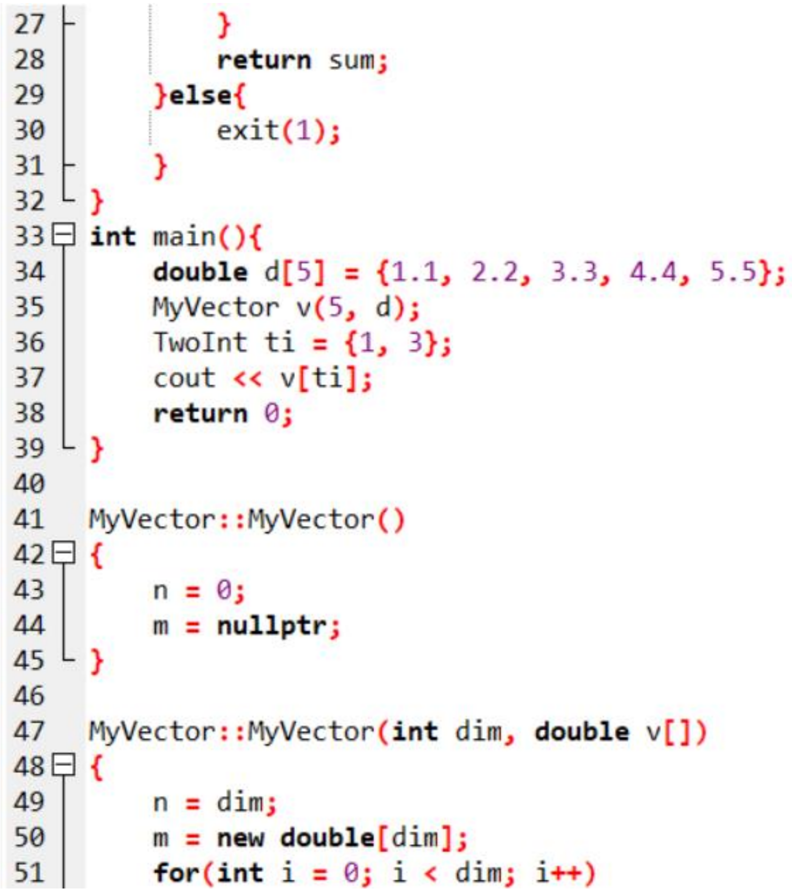
第二題

不可能，因為[]只能傳入一個參數，即使 overload，也有這個限制。

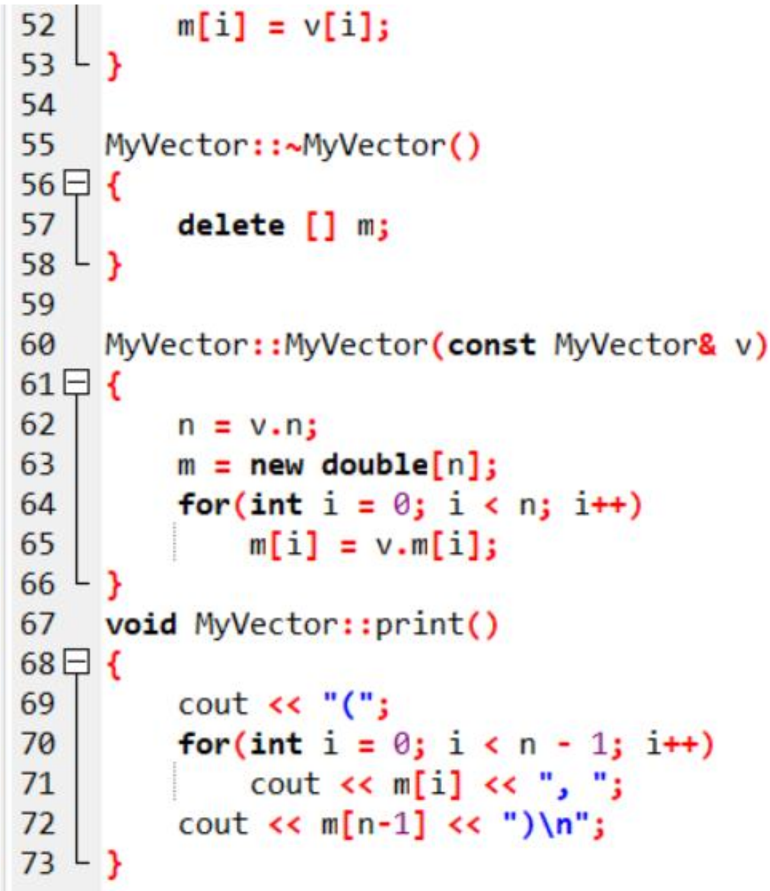
第三題



```cpp
#include<iostream>
#include<stdlib.h>
using namespace std;
struct TwoInt
{
    int i;
    int j;
};
class MyVector
{
private:
    int n;
    double* m;
public:
    MyVector();
    MyVector(int n, double m[]);
    MyVector(const MyVector& v);
    ~MyVector();
    void print();
    double operator[](struct TwoInt) const;
};
double MyVector::operator[](TwoInt p) const{
    if(p.i >= 0 && p.j > p.i && n-1 >= p.j){
        double sum = 0;
        for(int i = p.i;i <= p.j;i++){
            sum += m[i];
```

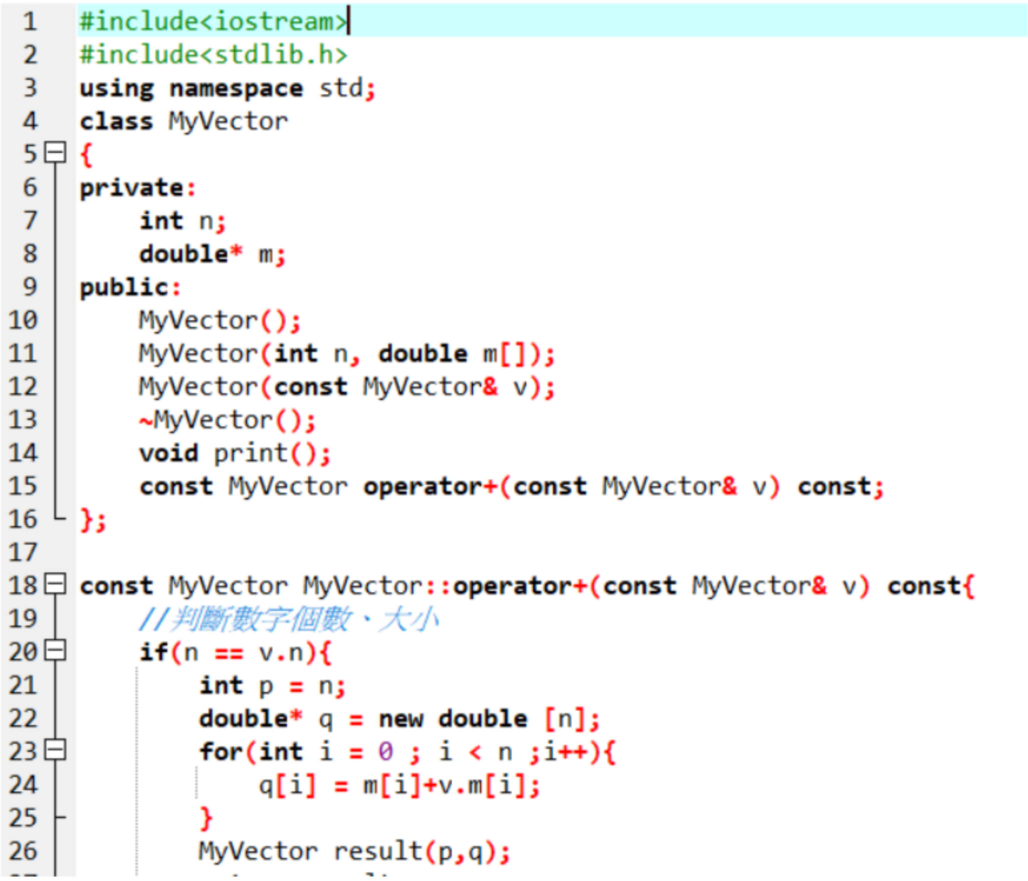


```cpp
        }
        return sum;
    }else{
        exit(1);
    }
}
int main(){
    double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
    MyVector v(5, d);
    TwoInt ti = {1, 3};
    cout << v[ti];
    return 0;
}

MyVector::MyVector()
{
    n = 0;
    m = nullptr;
}

MyVector::MyVector(int dim, double v[])
{
    n = dim;
    m = new double[dim];
    for(int i = 0; i < dim; i++)
```

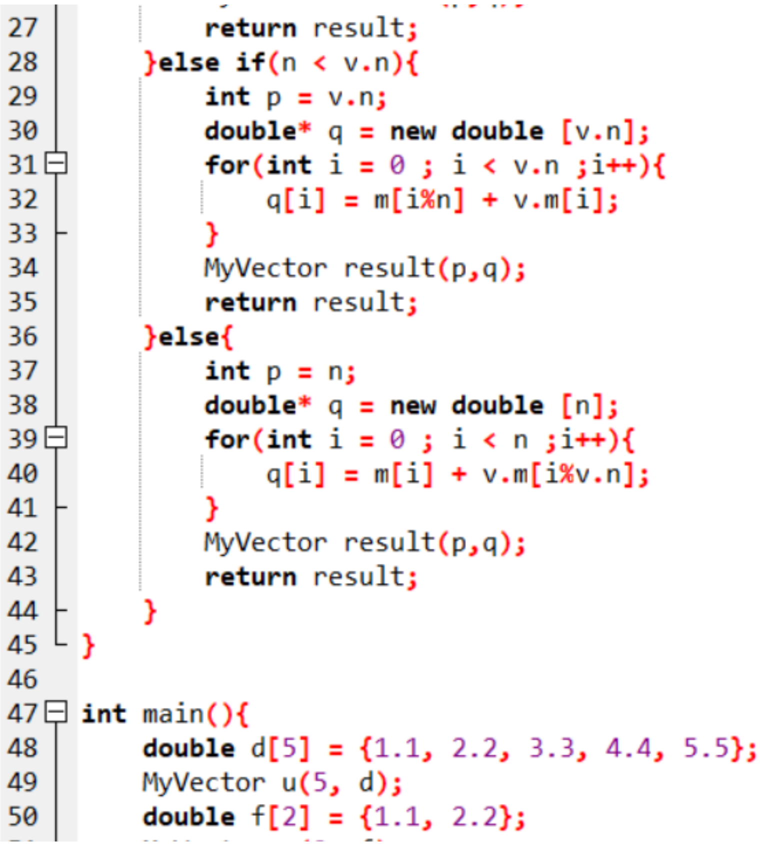


```
    m[i] = v[i];
}

MyVector::~MyVector()
{
    delete [] m;
}

MyVector::MyVector(const MyVector& v)
{
    n = v.n;
    m = new double[n];
    for(int i = 0; i < n; i++)
        m[i] = v.m[i];
}
void MyVector::print()
{
    cout << "(";
    for(int i = 0; i < n - 1; i++)
        cout << m[i] << ", ";
    cout << m[n-1] << ")\n";
}
```

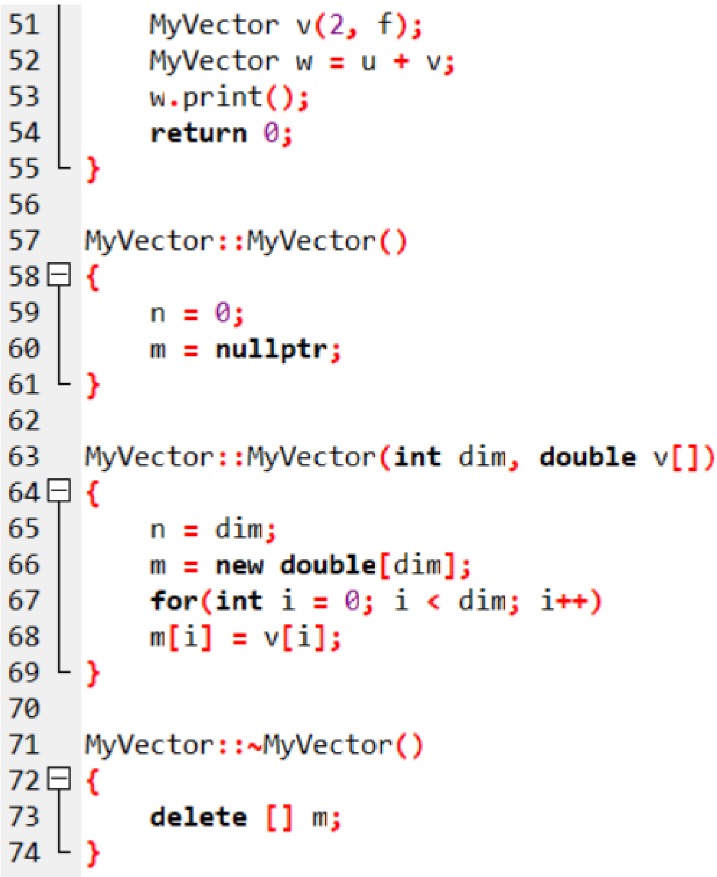
第四題



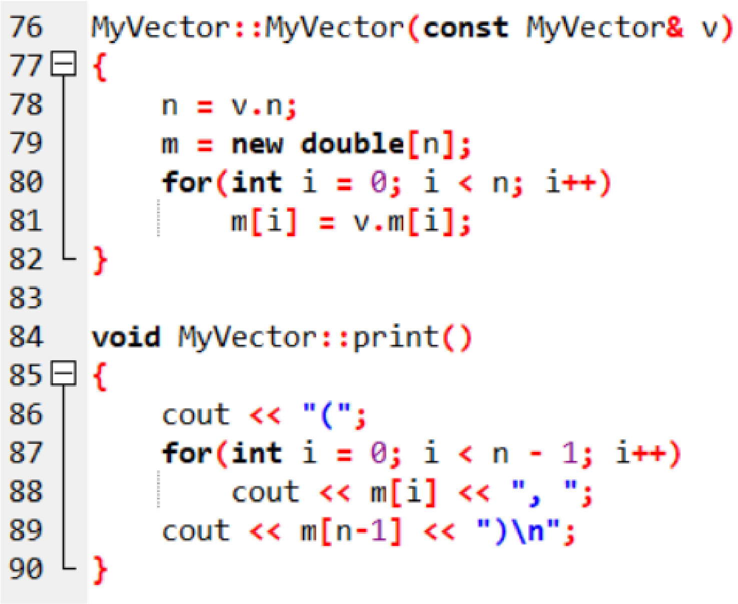
```
#include<iostream>
#include<stdlib.h>
using namespace std;
class MyVector
{
private:
    int n;
    double* m;
public:
    MyVector();
    MyVector(int n, double m[]);
    MyVector(const MyVector& v);
    ~MyVector();
    void print();
    const MyVector operator+(const MyVector& v) const;
};

const MyVector MyVector::operator+(const MyVector& v) const{
    //判斷數字個數、大小
    if(n == v.n){
        int p = n;
        double* q = new double [n];
        for(int i = 0 ; i < n ;i++){
            q[i] = m[i]+v.m[i];
        }
        MyVector result(p,q);
```



```
        return result;
    }else if(n < v.n){
        int p = v.n;
        double* q = new double [v.n];
        for(int i = 0 ; i < v.n ;i++){
            q[i] = m[i%n] + v.m[i];
        }
        MyVector result(p,q);
        return result;
    }else{
        int p = n;
        double* q = new double [n];
        for(int i = 0 ; i < n ;i++){
            q[i] = m[i] + v.m[i%v.n];
        }
        MyVector result(p,q);
        return result;
    }
}

int main(){
    double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
    MyVector u(5, d);
    double f[2] = {1.1, 2.2};
```

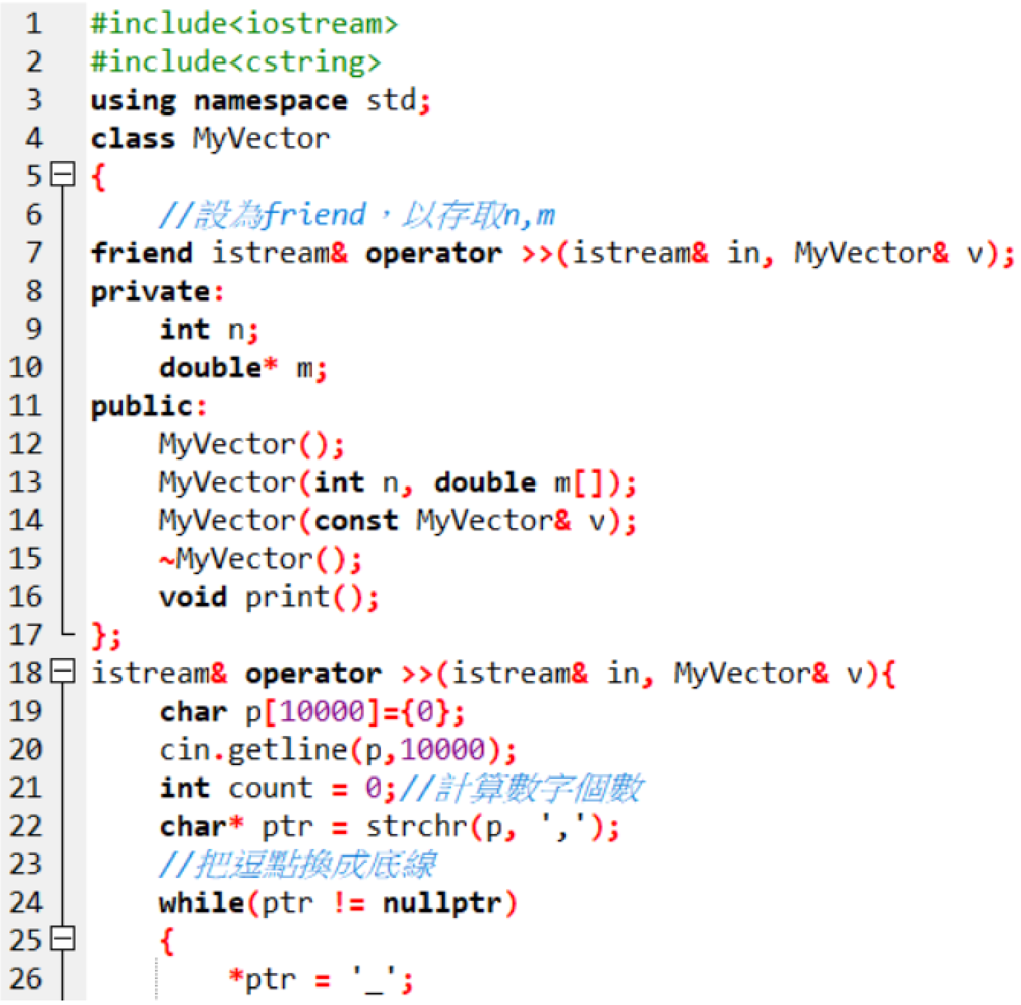


```
    MyVector v(2, f);
    MyVector w = u + v;
    w.print();
    return 0;
}

MyVector::MyVector()
{
    n = 0;
    m = nullptr;
}

MyVector::MyVector(int dim, double v[])
{
    n = dim;
    m = new double[dim];
    for(int i = 0; i < dim; i++)
    m[i] = v[i];
}

MyVector::~MyVector()
{
    delete [] m;
}
```

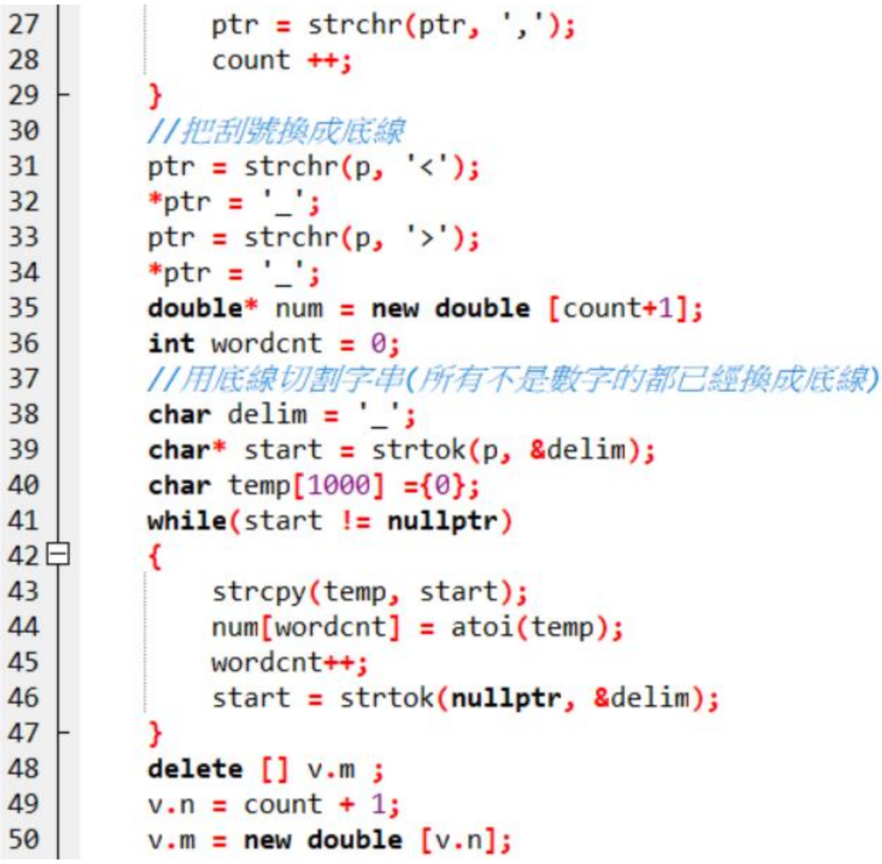


```
MyVector::MyVector(const MyVector& v)
{
    n = v.n;
    m = new double[n];
    for(int i = 0; i < n; i++)
        m[i] = v.m[i];
}

void MyVector::print()
{
    cout << "(";
    for(int i = 0; i < n - 1; i++)
        cout << m[i] << ", ";
    cout << m[n-1] << ")\n";
}
```

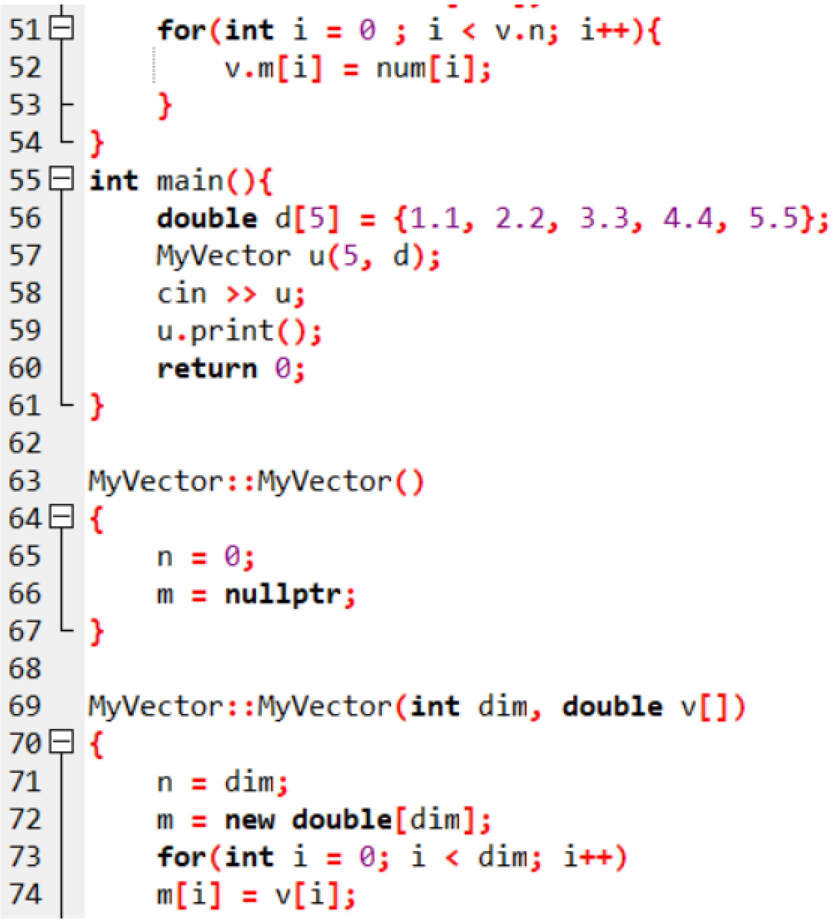
第五題



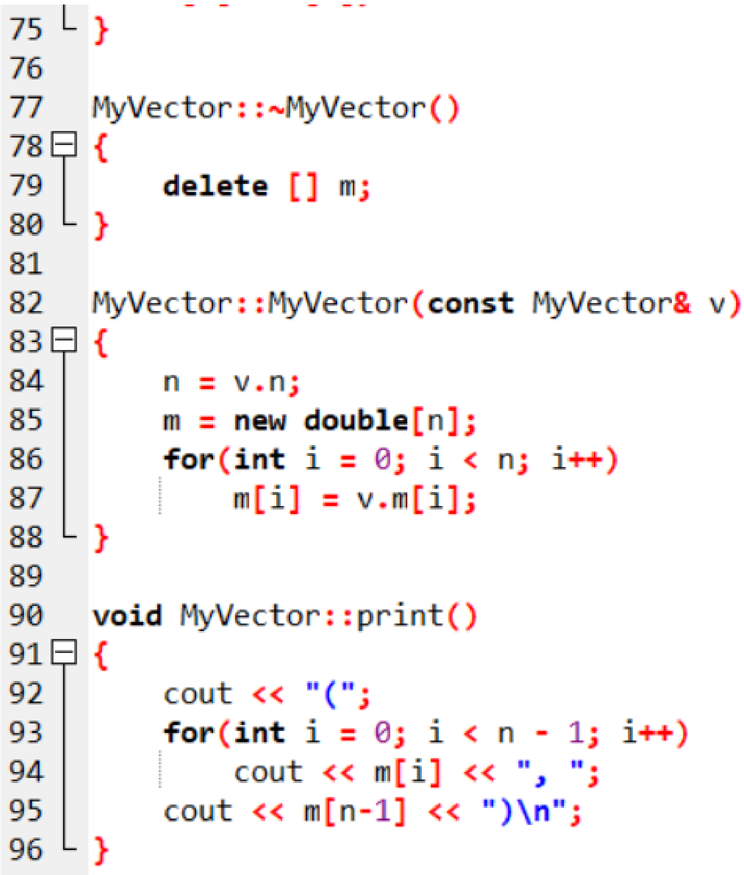
```
#include<iostream>
#include<cstring>
using namespace std;
class MyVector
{
    //設為friend，以存取n,m
friend istream& operator >>(istream& in, MyVector& v);
private:
    int n;
    double* m;
public:
    MyVector();
    MyVector(int n, double m[]);
    MyVector(const MyVector& v);
    ~MyVector();
    void print();
};
istream& operator >>(istream& in, MyVector& v){
    char p[10000]={0};
    cin.getline(p,10000);
    int count = 0;//計算數字個數
    char* ptr = strchr(p, ',');
    //把逗點換成底線
    while(ptr != nullptr)
    {
        *ptr = '_';
```



```
        ptr = strchr(ptr, ',');
        count ++;
    }
    //把刮號換成底線
    ptr = strchr(p, '<');
    *ptr = '_';
    ptr = strchr(p, '>');
    *ptr = '_';
    double* num = new double [count+1];
    int wordcnt = 0;
    //用底線切割字串(所有不是數字的都已經換成底線)
    char delim = '_';
    char* start = strtok(p, &delim);
    char temp[1000] ={0};
    while(start != nullptr)
    {
        strcpy(temp, start);
        num[wordcnt] = atoi(temp);
        wordcnt++;
        start = strtok(nullptr, &delim);
    }
    delete [] v.m ;
    v.n = count + 1;
    v.m = new double [v.n];
```



```
    for(int i = 0 ; i < v.n; i++){
        v.m[i] = num[i];
    }
}
int main(){
    double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
    MyVector u(5, d);
    cin >> u;
    u.print();
    return 0;
}

MyVector::MyVector()
{
    n = 0;
    m = nullptr;
}

MyVector::MyVector(int dim, double v[])
{
    n = dim;
    m = new double[dim];
    for(int i = 0; i < dim; i++)
    m[i] = v[i];
```



```
}

MyVector::~MyVector()
{
    delete [] m;
}

MyVector::MyVector(const MyVector& v)
{
    n = v.n;
    m = new double[n];
    for(int i = 0; i < n; i++)
        m[i] = v.m[i];
}

void MyVector::print()
{
    cout << "(";
    for(int i = 0; i < n - 1; i++)
        cout << m[i] << ", ";
    cout << m[n-1] << ")\n";
}
```