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){
        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];
        }
        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++)
```

```
52      m[i] = v[i];
53  }
54
55  MyVector::~MyVector()
56  {
57      delete [] m;
58  }
59
60  MyVector::MyVector(const MyVector& v)
61  {
62      n = v.n;
63      m = new double[n];
64      for(int i = 0; i < n; i++)
65          m[i] = v.m[i];
66  }
67  void MyVector::print()
68  {
69      cout << "(";
70      for(int i = 0; i < n - 1; i++)
71          cout << m[i] << ", ";
72      cout << m[n-1] << ")\n";
73  }
```

第四題

```
1   #include<iostream>
2   #include<stdlib.h>
3   using namespace std;
4   class MyVector
5   {
6   private:
7       int n;
8       double* m;
9   public:
10      MyVector();
11      MyVector(int n, double m[]);
12      MyVector(const MyVector& v);
13      ~MyVector();
14      void print();
15      const MyVector operator+(const MyVector& v) const;
16  };
17
18  const MyVector MyVector::operator+(const MyVector& v) const{
19      //判斷數字個數、大小
20      if(n == v.n){
21          int p = n;
22          double* q = new double [n];
23          for(int i = 0 ; i < n ;i++){
24              q[i] = m[i]+v.m[i];
25          }
26          MyVector result(p,q);
```

```
27          return result;
28      }else if(n < v.n){
29          int p = v.n;
30          double* q = new double [v.n];
31          for(int i = 0 ; i < v.n ;i++){
32              q[i] = m[i%n] + v.m[i];
33          }
34          MyVector result(p,q);
35          return result;
36      }else{
37          int p = n;
38          double* q = new double [n];
39          for(int i = 0 ; i < n ;i++){
40              q[i] = m[i] + v.m[i%v.n];
41          }
42          MyVector result(p,q);
43          return result;
44      }
45  }
46
47  int main(){
48      double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
49      MyVector u(5, d);
50      double f[2] = {1.1, 2.2};
51      MyVector v(2, f);
52      MyVector w = u + v;
53      w.print();
54      return 0;
55  }
56
57  MyVector::MyVector()
58  {
59      n = 0;
60      m = nullptr;
61  }
62
63  MyVector::MyVector(int dim, double v[])
64  {
65      n = dim;
66      m = new double[dim];
67      for(int i = 0; i < dim; i++)
68      m[i] = v[i];
69  }
70
71  MyVector::~MyVector()
72  {
73      delete [] m;
74  }
```

```
76  MyVector::MyVector(const MyVector& v)
77  {
78      n = v.n;
79      m = new double[n];
80      for(int i = 0; i < n; i++)
81          m[i] = v.m[i];
82  }
83
84  void MyVector::print()
85  {
86      cout << "(";
87      for(int i = 0; i < n - 1; i++)
88          cout << m[i] << ", ";
89      cout << m[n-1] << ")\n";
90  }
```

第五題

```
1   #include<iostream>
2   #include<cstring>
3   using namespace std;
4   class MyVector
5   {
6       //設為friend，以存取n,m
7   friend istream& operator >>(istream& in, MyVector& v);
8   private:
9       int n;
10      double* m;
11  public:
12      MyVector();
13      MyVector(int n, double m[]);
14      MyVector(const MyVector& v);
15      ~MyVector();
16      void print();
17  };
18  istream& operator >>(istream& in, MyVector& v){
19      char p[10000]={0};
20      cin.getline(p,10000);
21      int count = 0;//計算數字個數
22      char* ptr = strchr(p, ',');
23      //把逗點換成底線
24      while(ptr != nullptr)
25      {
26          *ptr = '_';
```

```
27          ptr = strchr(ptr, ',');
28          count ++;
29      }
30      //把刮號換成底線
31      ptr = strchr(p, '<');
32      *ptr = '_';
33      ptr = strchr(p, '>');
34      *ptr = '_';
35      double* num = new double [count+1];
36      int wordcnt = 0;
37      //用底線切割字串(所有不是數字的都已經換成底線)
38      char delim = '_';
39      char* start = strtok(p, &delim);
40      char temp[1000] ={0};
41      while(start != nullptr)
42      {
43          strcpy(temp, start);
44          num[wordcnt] = atoi(temp);
45          wordcnt++;
46          start = strtok(nullptr, &delim);
47      }
48      delete [] v.m ;
49      v.n = count + 1;
50      v.m = new double [v.n];
51      for(int i = 0 ; i < v.n; i++){
52          v.m[i] = num[i];
53      }
54  }
55  int main(){
56      double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
57      MyVector u(5, d);
58      cin >> u;
59      u.print();
60      return 0;
61  }
62
63  MyVector::MyVector()
64  {
65      n = 0;
66      m = nullptr;
67  }
68
69  MyVector::MyVector(int dim, double v[])
70  {
71      n = dim;
72      m = new double[dim];
73      for(int i = 0; i < dim; i++)
74      m[i] = v[i];
```

```
75  }
76
77  MyVector::~MyVector()
78  {
79      delete [] m;
80  }
81
82  MyVector::MyVector(const MyVector& v)
83  {
84      n = v.n;
85      m = new double[n];
86      for(int i = 0; i < n; i++)
87          m[i] = v.m[i];
88  }
89
90  void MyVector::print()
91  {
92      cout << "(";
93      for(int i = 0; i < n - 1; i++)
94          cout << m[i] << ", ";
95      cout << m[n-1] << ")\n";
96  }
```