

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

```
1  #include<iostream>
2  #include<cstring>
3  #include<stdlib.h>
4  using namespace std;
5  class MyVector
6  {
7  private:
8      int n;
9      double* m;
10 public:
11     MyVector();
12     MyVector(int n, double m[]);
13     MyVector(const MyVector& v);
14     ~MyVector();
15     void print();
16     double operator[](char* c) const;
17 };
18
19 double MyVector::operator[](char* c) const{
20     char* delim = ":";
21     char* ptr;
22     ptr = strtok(c,delim);
23     int num[2] = {0};
24     int count = 0;
25     //切開字串
26     while(ptr != nullptr){
27         num[count] = atoi(ptr);
28         count ++;
29         ptr = strtok(nullptr,delim);
30     }
31     //比較大小確定符合題目
32     if(num[0] >= 0 && num[1] > num[0] && n-1 >= num[1]){
33         double sum = 0;
34         for(int i = num[0];i <=num[1];i++){
35             sum += m[i];
36         }
37         return sum;
38     }else{
39         exit(1);
40     }
41 }
42 int main(){
43     double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
44     MyVector v(5, d);
45     char c[] = "1:3";
46     cout << v[c];
47     return 0;
48 }
49
50 MyVector::MyVector()
```

```

51 {
52     n = 0;
53     m = nullptr;
54 }
55
56 MyVector::MyVector(int dim, double v[])
57 {
58     n = dim;
59     m = new double[dim];
60     for(int i = 0; i < dim; i++)
61         m[i] = v[i];
62 }
63
64 MyVector::~MyVector()
65 {
66     delete [] m;
67 }
68
69 MyVector::MyVector(const MyVector& v)
70 {
71     n = v.n;
72     m = new double[n];
73     for(int i = 0; i < n; i++)
74         m[i] = v.m[i];
75 }
76
77 void MyVector::print()
78 {
79     cout << "(";
80     for(int i = 0; i < n - 1; i++)
81         cout << m[i] << ", ";
82     cout << m[n-1] << ")\n";
83 }
84

```

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

```

1  #include<iostream>
2  #include<stdlib.h>
3  using namespace std;
4  struct TwoInt
5  {
6      int i;
7      int j;
8  };
9  class MyVector
10 {
11 private:
12     int n;
13     double* m;
14 public:
15     MyVector();
16     MyVector(int n, double m[]);
17     MyVector(const MyVector& v);
18     ~MyVector();
19     void print();
20     double operator[](struct TwoInt) const;
21 };
22 double MyVector::operator[](TwoInt p) const{
23     if(p.i >= 0 && p.j > p.i && n-1 >= p.j){
24         double sum = 0;
25         for(int i = p.i; i <= p.j; i++){
26             sum += m[i];
27         }
28         return sum;
29     }else{
30         exit(1);
31     }
32 }
33 int main(){
34     double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
35     MyVector v(5, d);
36     TwoInt ti = {1, 3};
37     cout << v[ti];
38     return 0;
39 }
40
41 MyVector::MyVector()
42 {
43     n = 0;
44     m = nullptr;
45 }
46
47 MyVector::MyVector(int dim, double v[])
48 {
49     n = dim;
50     m = new double[dim];
51     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] = stod(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 return in;
55 }
56 int main(){
57     double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
58     MyVector u(5, d);
59     cin >> u;
60     u.print();
61     return 0;
62 }
63
64 MyVector::MyVector()
65 {
66     n = 0;
67     m = nullptr;
68 }
69
70 MyVector::MyVector(int dim, double v[])
71 {
72     n = dim;
73     m = new double[dim];
74     for(int i = 0; i < dim; i++)
75         m[i] = v[i];
76 }

```



```

77
78 MyVector::~MyVector()
79 {
80     delete [] m;
81 }
82
83 MyVector::MyVector(const MyVector& v)
84 {
85     n = v.n;
86     m = new double[n];
87     for(int i = 0; i < n; i++)
88         m[i] = v.m[i];
89 }
90
91 void MyVector::print()
92 {
93     cout << "(";
94     for(int i = 0; i < n - 1; i++)
95         cout << m[i] << ", ";
96     cout << m[n-1] << ")\n";
97 }

```