

Q01-a

```
1 #include <iostream>
2 #include <algorithm>
3 # include <vector>
4
5 using namespace std;
6
7 int main(void)
8 {
9     int n, k;
10    cout << "Please input an array a[n]!" << endl << "n = ";
11    cin >> n;
12    vector<int> vec;
13    cout << "Please input the array number:" << endl;
14    for(int i=0; i<n; i++)
15    {
16        cin >> k;
17        vec.push_back(k);
18    }
19    cout << "The number of occurences of '" << vec[0] << "' in the array is: ";
20    cout << count(vec.begin(), vec.end(), vec[0]) << endl;
21    return 0;
22 }
```

Q01-b

```
1 #include <iostream>
2 #include <cstdlib>
3
4 using namespace std;
5
6 int m, n;
7 int arr[1005][1005];
8 int ans[1005][1005];
9
10 void matrix_trans(int m, int n)
11 {
12     int x;
13     if(m>=n)
14         x=m;
15     else
16         x=n;
17     for(int i=1; i<x+1; i++)
18     {
19         for(int j=1; j<x+1; j++)
20         {
21             ans[i][j]=arr[j][i];
22         }
23     }
24     for(int i=1; i<n+1; i++)
25     {
26         for(int j=1; j<m+1; j++)
27         {
28             cout << ans[i][j] << " ";
29         }
30         cout << endl;
31     }
32 }
33
34 int main(void)
35 {
36     cout << "Please decide the size of an 'm*n' matrix:" << endl;
37     cout << "m = ";
38     cin >> m;
39     cout << "n = ";
40     cin >> n;
41     cout << "Please input the contain number of the matrix:" << endl;
42     for(int i=1; i<m+1; i++)
43     {
44         for(int j=1; j<n+1; j++)
45         {
46             cin >> arr[i][j];
47         }
48     }
49     matrix_trans(m, n);
50     return 0;
51 }
52
```

Q02

$$2. (a) \sum_{\lambda=0}^n \lambda^2 = \Theta(n^3)$$

$$0^2 + 1^2 + 2^2 + \dots + n^2 = \underbrace{1^2 + 2^2 + 3^2 + \dots + n^2}_n \leq \underbrace{n^1 + n^1 + \dots + n^1}_n = n \cdot n^1 = n^2$$

$$\therefore \forall n \in \mathbb{R}, n > 0$$

$$0 \cdot n^3 \leq \sum_{\lambda=0}^n \lambda^2 \leq n^3$$

$$\therefore \sum_{\lambda=0}^n \lambda^2 = \Theta(n^3)_{\#}$$

$$\left(\lim_{n \rightarrow \infty} \frac{\frac{1}{6} n(n+1)(2n+1)}{n^3} \right)$$

$$= \lim_{n \rightarrow \infty} \frac{\frac{1}{6} (n^3 + n)(2n+1)}{n^3} = \frac{1}{3} \neq$$

$$(b) n! = O(n^n)$$

$$\forall n, \text{ when } n > 0$$

$$n! = \underbrace{1 \times 2 \times \dots \times n}_n \leq \underbrace{n \times n \times \dots \times n}_n = n^n$$

$$\left(\lim_{n \rightarrow \infty} \frac{n!}{n^n} = 0 \neq \right)$$

$$\therefore n! = O(n^n)_{\#}$$

Q03

3. (a) $10n^2 + 9$

$$\forall n, n \geq 3, 10n^2 + 9 \leq 11n^2$$

$$\therefore 10n^2 + 9 = O(n^2) \#$$

$$\text{If } 10n^2 + 9 = O(n)$$

Then suppose $a \in \mathbb{R}, b \in \mathbb{R}$

Let $10n^2 + 9 \leq an + b$ (By the definition of big oh)

$$10n^2 + 9 \leq an + b$$

$$\Rightarrow \underline{10n^2 - an + (9 - b)} \leq 0$$

when this function has to be negative in constant

the coefficient of n^2 should be negative,

but $10 > 0$, so the suppose failed.

Therefore, $10n^2 + 9 = O(n)$ is wrong #

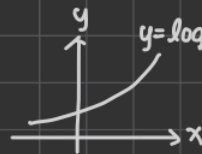
(b) If $\frac{n^2}{\log n} = \Theta(n^2)$

Suppose $c_1, c_2 \in \mathbb{R}$

Let $g(n) = n^2$

$$c_1 \cdot n^2 \leq \frac{n^2}{\log n} \leq c_2 \cdot n^2$$

$$c_1 \leq \frac{1}{\log n} \leq c_2$$



$y = \log x \rightarrow$ there's no absolute maximum / $\lim_{x \rightarrow \infty} (\log n)$
so we can't find c_2 that always
satisfy $\frac{1}{\log n} < c_2 \ (\forall n \in \mathbb{R})$

Therefore, $\frac{n^2}{\log n} = \Theta(n^2)$ is wrong #

Q04

```
1  #include <iostream>
2
3  using namespace std;
4
5  class Complex
6  {
7  private:
8      int real, imagine;
9  public:
10     Complex(int a=0, int b=0)
11     {
12         real = a;
13         imagine = b;
14     }
15
16     void print()
17     {
18         cout << real << " + " << imagine << "i";
19     }
20 };
21
22 int main(int argc, const char * argv[])
23 {
24     int a, b;
25     cout << "Please input a complex number ai + b" << endl;
26     cout << "The real part of the complex number a: " << endl;
27     cin >> a;
28     cout << "The imaginary part of the complex number b:" << endl;
29     cin >> b;
30     cout << "Default number: " ;
31     Complex Default(0,0);
32     Default.print();
33     cout << endl << "Complex number: " ;
34     Complex Complex(a, b);
35     Complex.print();
36     return 0;
37 }
```

Q05

```
1  #include <iostream>
2
3  using namespace std;
4
5  int x[5], y[5], z[5];
6
7  class Quadratic
8  {
9  private:
10     int a_sum=0, b_sum=0, c_sum=0;
11 public:
12     Quadratic()
13     {
14         a_sum = x[1] + x[2];
15         b_sum = y[1] + y[2];
16         c_sum = z[1] + z[2];
17     }
18
19     void print()
20     {
21         cout << a_sum << "x^2 + " << b_sum << "x + " << c_sum;
22         cout << endl;
23     }
24 };
25
26
27 int main(void)
28 {
29     cout << "Now please enter polynomials 1: ax^2 + bx + c:" << endl;
30     cout << "a = ";
31     cin >> x[1];
32     cout << "b = ";
33     cin >> y[1];
34     cout << "c = ";
35     cin >> z[1];
36     cout << "Now please enter polynomials 2: ax^2 + bx + c:" << endl;
37     cout << "a = ";
38     cin >> x[2];
39     cout << "b = ";
40     cin >> y[2];
41     cout << "c = ";
42     cin >> z[2];
43     cout << "The sum of the polynomials is:" << endl;
44     Quadratic test;
45     test.print();
46
47     return 0;
48 }
49
```

Q06

```
1  #include<iostream>
2
3  using namespace std;
4
5  struct a
6  {
7      int n;
8      a* next;
9  };
10
11 class bag
12 {
13 public:
14     a* f, * l;
15
16     bag()
17     {
18         f=new a;
19         f->next=NULL;
20         l=f;
21     }
22
23     void add(int n)
24     {
25         a* temp=new a;
26         temp->n=n;
27         temp->next=NULL;
28         l->next=temp;
29         l=temp;
30         a* current=f;
31         cout << "Number:";
32         while(current->next != NULL)
33         {
34             cout << current->next->n << " ";
35             current=current->next;
36         }
37         cout << endl;
38     }
39
40     void initialize()
41     {
42         f=new a;
43         f->next=NULL;
44         l=f;
45     }
46
47     void del(int n)
48     {
49         int something=1;
50         int time=0;
51         a* current=f;
52         if(current == NULL)
53         {
54             cout << n << " isn't in the bag." << endl;
55         }
56         else
57         {
58             while(current->next != NULL)
59             {
60                 if(current->next->n == n)
61                 {
62                     something=0;
63                     a* temp = current->next;
64                     current->next = temp->next;
65                     delete(temp);
66                     time++;
67                     cout << "The bag had delete " << time << " time of the number " << n << ":";
68                     a* temp_2=f;
```

```

69         while(temp_2->next != NULL)
70         {
71             cout << temp_2->next->n << " ";
72             temp_2 = temp_2->next;
73         }
74         cout << endl;
75     }
76     else
77     {
78         current = current->next;
79     }
80 }
81 if(something==1)
82 {
83     cout << "No number " << n << " int the bag." << endl;
84 }
85 }
86 cout << endl;
87 }
88 };
89
90 class queue :public bag
91 {
92 public:
93     queue()
94     {
95         this->f=new a;
96         this->f->next=NULL;
97         this->l=this->f;
98     }
99     void del_queue()
100    {
101        a* temp=f;
102        if(temp->next == NULL)
103        {
104            cout << "The queue has been empty!" << endl << endl;
105        }
106        else
107        {
108            cout << "Delete num " << temp->next->n << " from the first place of this queue." << endl;
109            f=temp->next;
110            delete(temp);
111            a* current=f;
112            cout << "The queue now has the number:";
113            while(current->next != NULL)
114            {
115                cout << current->next->n << " ";
116                current = current->next;
117            }
118            cout << endl;
119        }
120    }
121 };
122
123 int main()
124 {
125     int n;
126     cout << "Make a bag, input 1. Make a queue, input 2." << endl;
127     cin >> n;
128     if (n == 1)
129     {
130         bag bag;
131         int something=1;
132         while(something == 1)
133         {
134             int m;
135             cout << "Add number, input 1.Delete number, input 2.Stop this bag, input 3." << endl;
136             cin >> m;
137             if(m == 1)
138             {
139                 int o;
140                 cout << "Add a new number = ";
141                 cin >> o;
142                 bag.add(o);
143             }
144             else if(m == 2)
145             {
146                 int p;
147                 cout << "Delete a number = ";
148                 cin >> p;
149                 bag.del(p);
150             }
151             else if(m == 3)
152                 something=0;
153             else
154                 cout << "Error!";
155         }
156     }

```



```
157     else if(n == 2)
158     {
159         queue queue;
160         int something_2=1;
161         while(something_2 == 1)
162         {
163             int q;
164             cout << "Input 1 to add a new number. Input 2 to delete number. Input 3 to stop." << endl;
165             cin >> q;
166             if (q == 1)
167             {
168                 int r;
169                 cout << "Add a new number = ";
170                 cin >> r;
171                 queue.add(r);
172             }
173             else if(q == 2)
174                 queue.del_queue();
175             else if(q == 3)
176                 something_2=0;
177             else
178                 cout << "Error!";
179         }
180     }
181     else
182     {
183         cout << "Error!";
184     }
185     return 0;
186 }
187
```

Q07

7. $A * B * C$

→ prefix form: $**ABC\#$

Q08

```
1  #include<iostream>
2  #include<vector>
3  #include<cstring>
4
5  using namespace std;
6
7  class A
8  {
9  public:
10     int row, col, val;
11     A(int x = -1, int col = -1, int val = -1) : row(row), col(col), val(val) {}
12 };
13
14 class Matrix
15 {
16 public:
17     Matrix(){}
18     Matrix(int row_num, int col_num, int r) : row_num(row_num), col_num(col_num), r(r) {}
19     Matrix(const Matrix& other);
20     Matrix operator*(Matrix& other);
21     void set_M();
22     void set_vec();
23     void display();
24
25     int row_num;
26     int col_num;
27     int r;
28     int M[100][100];
29     vector<A> vec;
30 };
31
32 Matrix::Matrix(const Matrix& copy)
33 {
34     row_num = copy.row_num;
35     col_num = copy.col_num;
36     r = copy.r;
37     vec = copy.vec;
38     for(int i=0; i<row_num; i++)
39     {
40         for(int j=0; j<col_num; j++)
41         {
42             M[i][j]=copy.M[i][j];
43         }
44     }
45 }
46
47 void set_input(Matrix& K)
48 {
49     cin >> K.row_num >> K.col_num >> K.r;
50     for(int i=0; i<K.r; i++)
51     {
52         int row, col, val;
53         cin >> row >> col >> val;
54         A a(row, col, val);
55         K.vec.push_back(a);
56     }
57     K.set_M();
58 }
59
60 void Matrix::set_M()
61 {
62     memset(M, 0, sizeof(M));
63     for(int i=0; i<r; i++)
64         M[vec[i].row][vec[i].col]=vec[i].val;
65 }
66
```

```

67 void Matrix::set_vec()
68 {
69     int count=0;
70     for(int i=0; i<row_num; i++)
71     {
72         for(int j=0; j<col_num; j++)
73         {
74             if (M[i][j] != 0)
75             {
76                 A element(i, j, M[i][j]);
77                 vec.push_back(element);
78                 count++;
79             }
80         }
81     }
82     r = count;
83 }
84
85 void Matrix::display()
86 {
87     for(int i=0; i<row_num; i++)
88     {
89         for(int j=0; j<col_num; j++)
90             cout << M[i][j] << " ";
91         cout << endl;
92     }
93 }
94
95 Matrix Matrix::operator*(Matrix& other)
96 {
97     Matrix mul_f(row_num, other.col_num, 0);
98
99     if(other.col_num*r <= row_num*other.r)
100     {
101         for(int i=0; i<r; i++)
102         {
103             for(int j=0; j<other.col_num; j++)
104                 mul_f.M[vec[i].row][j] += vec[i].val*other.M[vec[i].col][j];
105         }
106     }
107     else
108     {
109         for(int i=0; i<other.r; i++)
110         {
111             for(int j=0; j<row_num; j++)
112                 mul_f.M[j][other.vec[i].col] += other.vec[i].val*M[j][other.vec[i].row];
113         }
114     }
115     mul_f.set_vec();
116     return mul_f;
117 }
118
119 int main()
120 {
121     Matrix Sample;
122     Matrix Sample_2;
123     set_input(Sample);
124     set_input(Sample_2);
125
126     Matrix copy_SM=Sample;
127     cout << "copy result: " << endl;
128     copy_SM.display();
129     return 0;
130 }

```

Q09

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <stdio.h>
4
5  using namespace std;
6
7  int main(void)
8  {
9      int arr[6];
10     int i;
11     for(i=1; i<6; i++)
12     {
13         //scanf("%d", arr[i]);
14         cin >> arr[i];
15         for(int j=1; j<i; j++)
16         {
17             int key=0;
18             if(arr[i]<arr[j])
19             {
20                 key=1;
21                 int temp=arr[j];
22                 arr[j]=arr[i];
23                 for(int k=i; k>j; k--)
24                 {
25                     arr[k]=arr[k-1];
26                     if(k==j+1)
27                     {
28                         arr[k]=temp;
29                         break;
30                     }
31                 }
32             }
33             if(key==1) break;
34         }
35     }
36     for(i=1; i<6; i++)
37     {
38         //printf("%d ", arr[i]);
39         cout << arr[i] << " ";
40     }
41     return 0;
42 }
```

Q10

```
1 #include <iostream>|
2 #include <string>
3
4 using namespace std;
5
6 void pal(string a)
7 {
8     string b;
9     unsigned long int n;
10    n=a.size();
11    for(unsigned long int i=0, j=n-1; i<n; i++, j--)
12    {
13        b[j]=a[i];
14    }
15    int key=0;
16    for(int i=0; i<n; i++)
17    {
18        if(a[i] != b[i])
19        {
20            key=1;
21            break;
22        }
23    }
24    if(key==1)
25    {
26        cout << "This isn't a palindrome.";
27    }
28    else
29        cout << "This is a palindrome.";
30 }
31
32 int main(void)
33 {
34     string a, b;
35     cout << "Please type a word you'll like to test:" << endl;
36     cin >> a;
37     pal(a);
38     return 0;
39 }
40
```