3.提升代码能力

1、斐波那契数列

```
#include <iostream>
using namespace std;
const int mod = 1e9 + 7;
int f[100005];
int main()
{
    int n;
    cin >> n;
    f[1] = 1;
    f[2] = 1;
    for (int i = 3; i <= n; i++)
    {
        f[i] = (f[i-1] + f[i-2]) % mod;
    }
    cout << f[n] << endl;
    return 0;
}
```

2、旋转矩阵

```
int num[205][205];
int main()
{
    int n,m;
    cin >> n >> m;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m;j++)
            cin >> num[i][j];
    }

    for(int i=0; i < m; i++)
    {
        for (int j = 0;j < n; j++)
        {
            if (j!= n-1)
                cout << num[n-1-j][i] << " ";
            else
                cout << num[n-1-j][i] << endl;
        }
```

```
    }
    return 0;
}
```

3、最大子矩阵（暴力）

```
int A[55][55];

int main()
{
    int n,m,ans;
    cin >> n >> m;
    ans = -1005;
    for (int i = 0 ;i < n;i++)
    {
        for (int j = 0; j < m;j++)
        {
            cin >> A[i][j];
        }
    }

     for (int i = 0; i < n; i++)    //i,j,k,l 上下左右四条边界
     {
        for (int j = i; j < n; j++)
        {
            for (int k = 0; k < m;k++)
            {
                for (int l = k; l < m; l++)
                {
                    int tmp = 0;
                    for (int p = i; p <= j ;p++)
                    {
                        for (int q = k; q <= l; q++)
                        {
                            tmp += A[p][q];
                         }
                         if (tmp > ans)
                          ans = tmp;
                     }
                 }
              }
          }
     }
     cout << ans <<endl;
    return 0;
}
```

4、去重

```
int num[105],ans[105];
int main()
```

```
{
    int n,m=0;
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cin >> num[i];
    }
    sort(num, num+n);
    for (int i = 0; i < n; i++)
    {
        if ( i!= 0 && num[i] != num[i-1])
        {
            ans[m++] = num[i-1];
        }
    }
    ans[m++] = num[n-1];
    cout << m <<endl;
    for (int i =0 ; i < m; i++)
    {
        cout << ans[i] <<" ";
    }
    return 0;
}
```

5、进制转换

```
char ans[105];
int main()
{
    int N,R,m,now;
    cin >> N >> R;
    if ( N < 0)
    {
        cout << "-";
        N = - N;
    }
    while ( N )
    {
        now = N % R;
        if (now <= 9)
        {
            ans[m++] = '0' + now;
        }
        else
        {
            ans[m++] = 'A' +now -10;
        }
        N /= R;
    }
    if ( !m)
    {
```

```
        cout << 0 ;
    }
    for (int i = m - 1; i>=0; i--)
    {
        cout << ans[i];
    }
    return 0;
}
```

6、回文数

```
int num[10005];
int digit[10005];
bool judge(int x)
{
    int cnt = 0;
    while(x)
    {
        digit[cnt++] = x % 10;
        x /= 10;
    }
    for(int i = 0 ; i < cnt/2; i++)
    {
        if (digit[i] != digit[cnt-1-i])
            return false;
    }
    return true;
}

int rev(int x)
{
    int ret = 0;
    while(x)
    {
        ret = ret * 10 + x % 10;
        x /= 10;
    }
    return ret;
}
int main()
{
    int n,m;
    cin >> n;
    m = 0;
    num[m++] = n;
    while (!judge(n))
    {
         n += rev(n);
         num[m++] = n;
    }
    cout << m - 1 <<endl;
```

```
    for (int i = 0; i < m; i++)
    {
        if ( i != m - 1)
            cout << num [i] << "-->";
        else
            cout << num [i] << endl;
    }
    return 0;
}
```

6、机器人

```
#include <iostream>
#include <algorithm>
using namespace std;

int dx[4] = {0,-1,0,1};
int dy[4] = {1,0,-1,0};
char op[15];
int main()
{
    int n,d,x,nowx=0,nowy=0;
    cin >> n;
    d = 3;
    for (int i = 0; i < n; i++)
    {
        cin >> op >> x;
        if (op[0] == 'b')
        {
            d = (d+2)%4;
        }
        else if (op[0] == 'l')
        {
            d = (d+1)%4;
        }
        else if (op[0] == 'r')
        {
            d = (d+3)%4;
        }
        nowx += dx[d] * x;
        nowy += dy[d] * x;
    }
    cout << nowx << nowy;
    return 0;
}
```