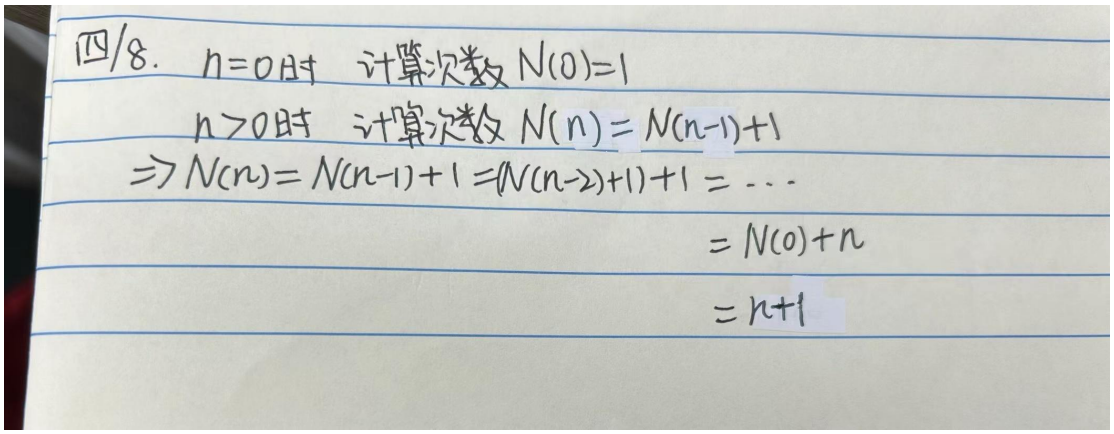
四/8



五/2

```
#include<stack>
#include "List.h"
using namespace std;
bool if_center_symmetric(List<int> a) {
    int len = a.Length();
    stack<int> b;
    if (len % 2 == 0) {
        for (int i = 0;i < len / 2;i++) {
            int x;
            a.getData(i + 1, x);
            b.push(x);
        }
        for (int i = len / 2;i < len;i++) {
            int x;
            a.getData(i + 1, x);
            if (x != b.top()) return false;
            else b.pop();
        }
        return true;
    }
    else {
        for (int i = 0;i < (len - 1) / 2;i++) {
            int x;
            a.getData(i + 1, x);
            b.push(x);
        }
        for (int i = (len + 1) / 2;i < len;i++) {
            int x;
            a.getData(i + 1, x);
            if (x != b.top()) return false;
            else b.pop();
        }
        return true;
    }
}
```

五/8

```cpp
#include "SeqQueue.h"
using namespace std;
int max_fibonacci(int max, int k) {
    SeqQueue<int> f;
    int a = 0, b = 1;
    while (a < max) {
        if (f.getSize() != k) {
            f.EnQueue(a);
        }
        else {
            int x;
            f.DeQueue(x);
            f.EnQueue(a);
        }
        int c = a + b;
        a = b;
        b = c;
    }
    return a;
}
```

五/10

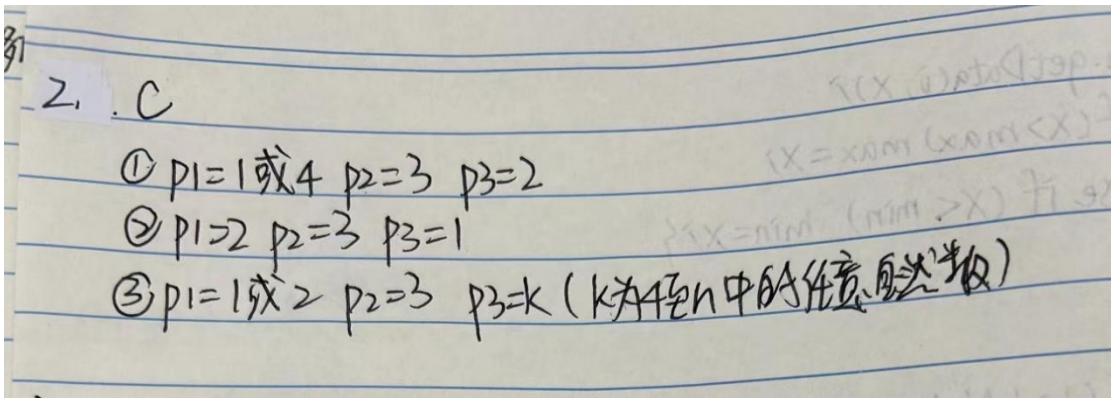
```cpp
#include <vector>
#include <stdlib.h>
#include<iostream>
using namespace std;
vector<vector<int>> num_combination(int n, int m) {
    vector<vector<int>> a;
    if (n == m) {
        vector<int> c;
        for (int i = n;i != 0;i--) {
            c.push_back(i);
        }
        a.push_back(c);
        return a;
    }
    if (m == 1) {
        for (int i = n;i != 0;i--) {
            vector<int> c;
            c.push_back(i);
            a.push_back(c);
        }
        return a;
    }
    else if(n != m && m != 1) {
        for (int i = n;i != m - 1;i--) {
            vector<vector<int>>d;
            if (i != m) {
                d = num_combination(i - 1, m - 1);
                int len = d.size();
                for (int j = 0;j < len;j++) {
                    vector<int> e = d[j];
                    d[j][0] = i;
                    for (int k = 1;k < d[j].size();k++) {
                        d[j][k] = e[k - 1];
                    }
                    int q = d[j].size();
                    d[j].push_back(e[q - 1]);
                }
            }
            if (i == m) d = num_combination(i, m);
            for (int p = 0;p < d.size();p++) {
                a.push_back(d[p]);
            }
        }
        return a;
    }
}
```

1.

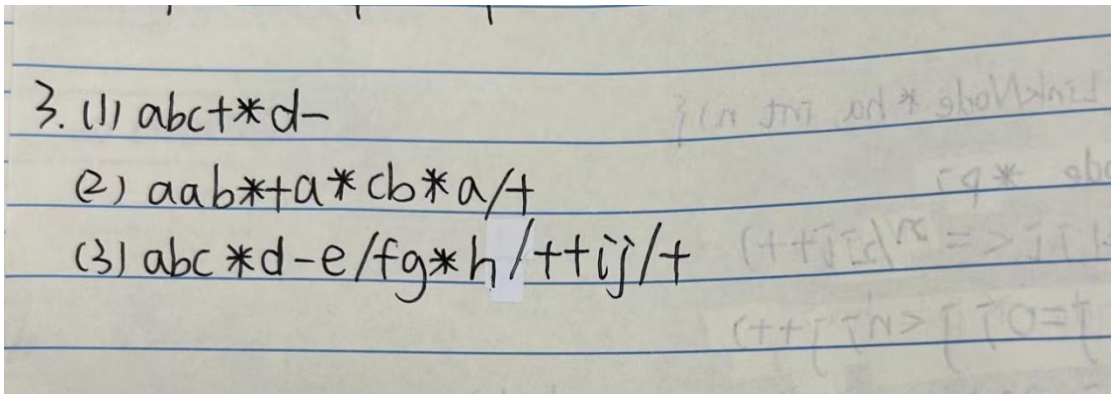
```cpp
#include <list>
#include <stack>
#include <iostream>
using namespace std;
bool if_trans(list<int> a, list<int> b) {
    if (a.size() != b.size()) return false;
    stack<int>c;
    int len = b.size();
    for (int i = 0;i < len;i++)
    {
        if (c.empty() != 0) {
            c.push(a.front());
            a.pop_front();
        }
        while (c.top() != b.front()&& a.empty() == 0) {
            c.push(a.front());
            a.pop_front();
        }
        if (c.top() == b.front()) {
            c.pop();
            b.pop_front();
        }
        else {
            cout << "no" << endl;
            return false;
        }
    }
    cout << "yes" << endl;
    return true;
}

int main(){
    list<int>a, b, c;
    for (int i = 1;i != 7;i++) {
        a.push_back(i);
    }
    b.push_back(3);b.push_back(2);b.push_back(5);b.push_back(6);b.push_back(4);b.push_back(1);
    if_trans(a, b);
    c.push_back(1);c.push_back(5);c.push_back(4);c.push_back(6);c.push_back(2);c.push_back(3);
    if_trans(a, c);
    return 0;
}
```

2.



3.



补充题 4 同五/10