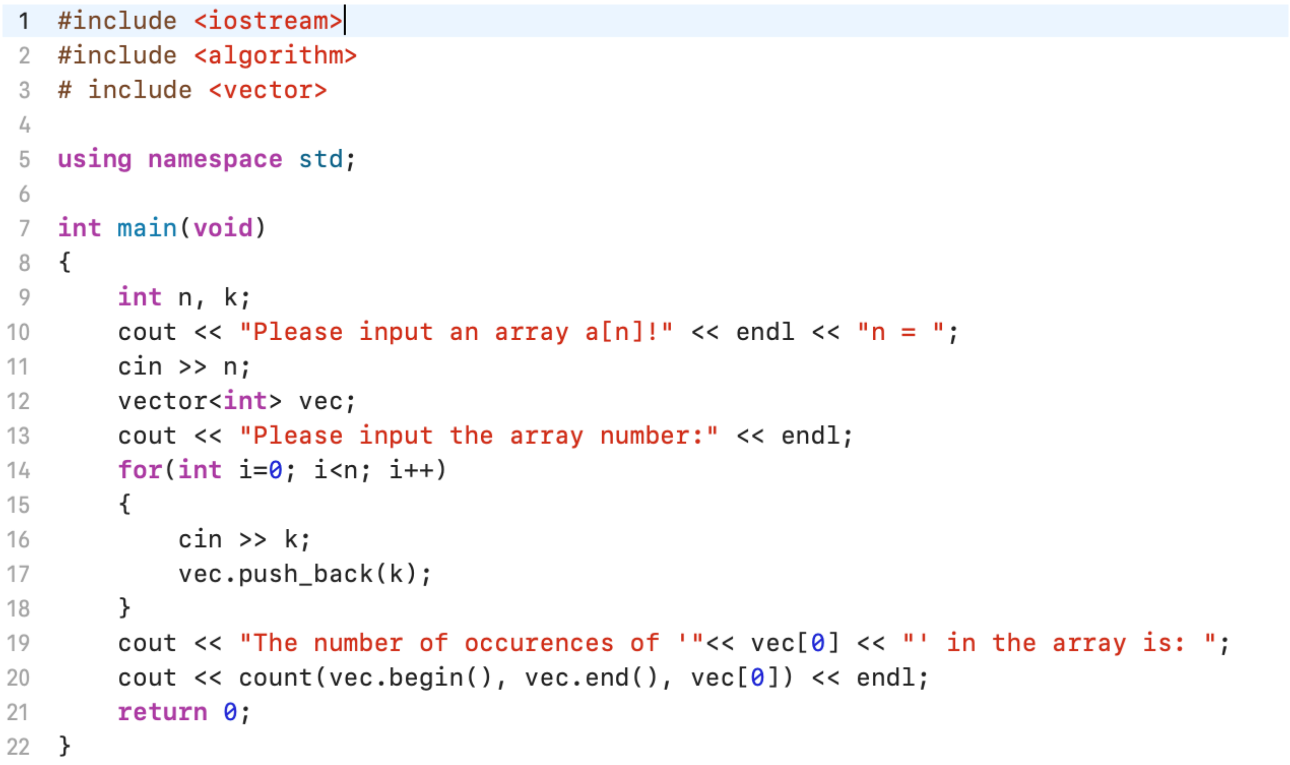
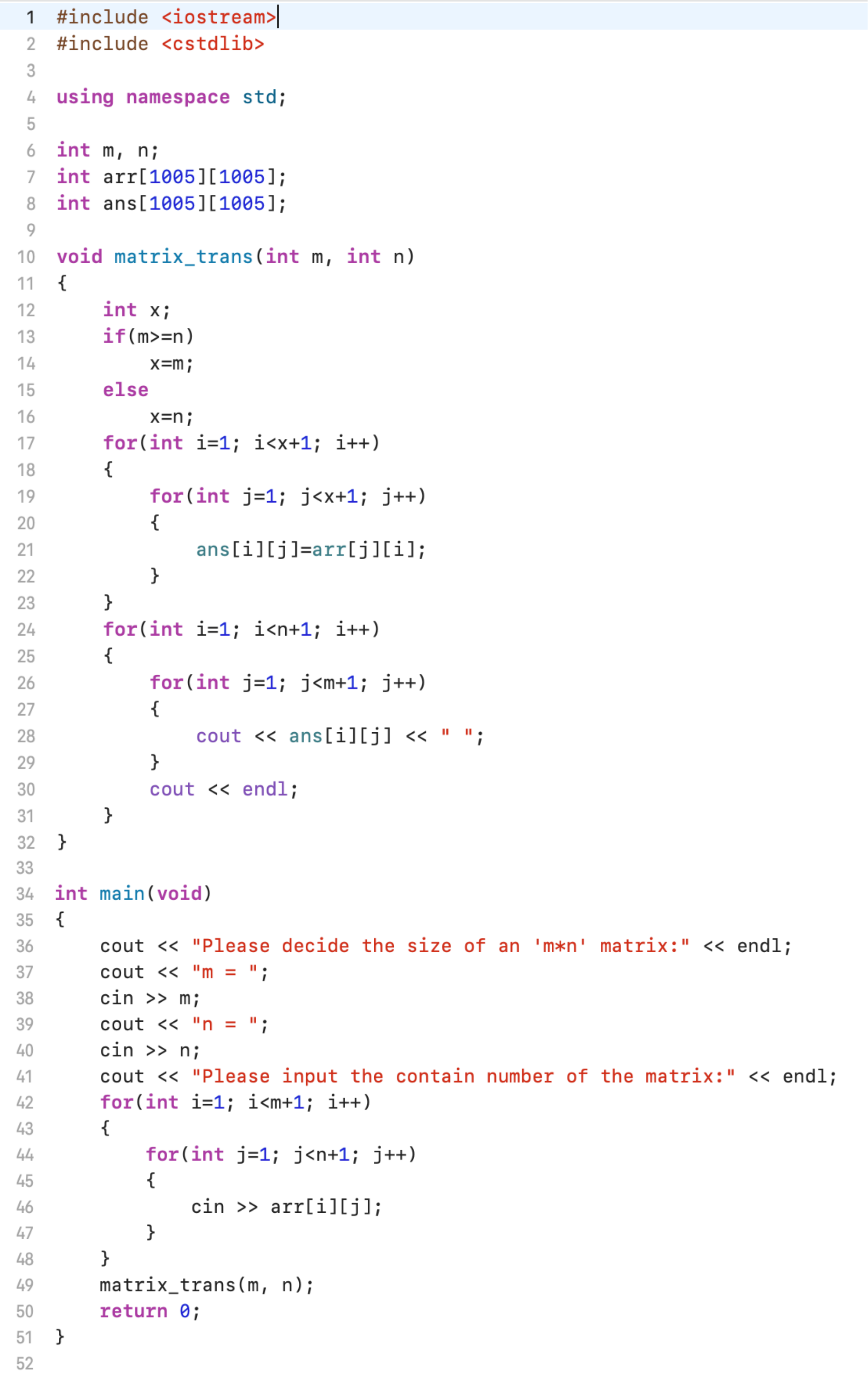
Q01–a



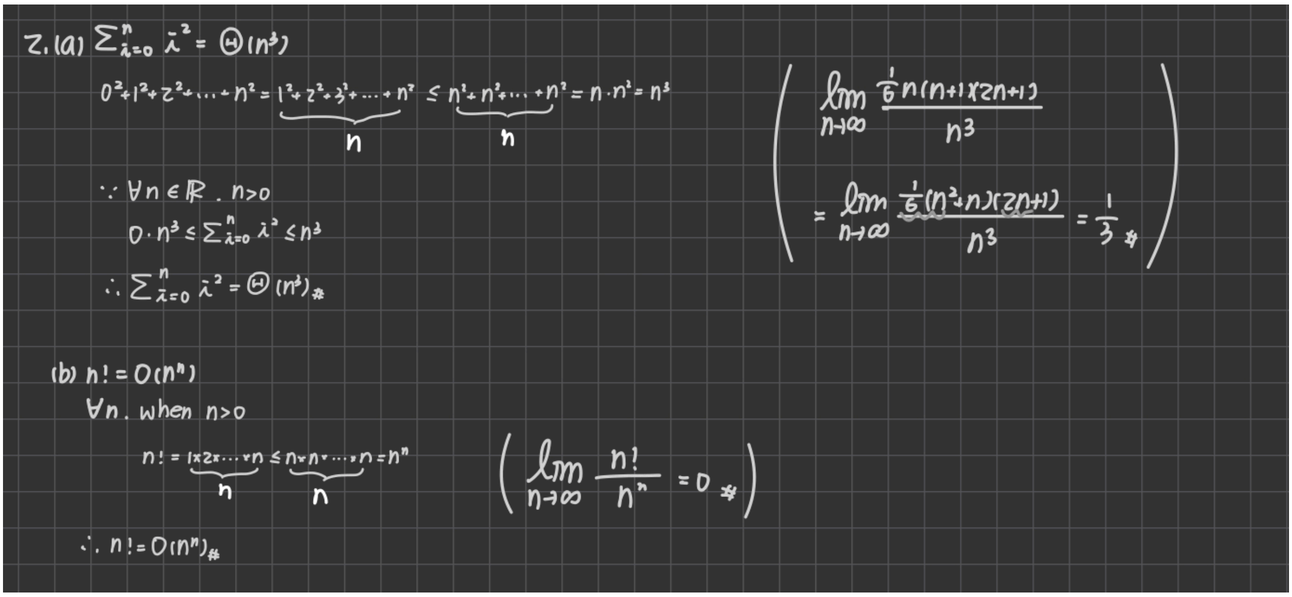
```cpp
#include <iostream>
#include <algorithm>
# include <vector>

using namespace std;

int main(void)
{
    int n, k;
    cout << "Please input an array a[n]!" << endl << "n = ";
    cin >> n;
    vector<int> vec;
    cout << "Please input the array number:" << endl;
    for(int i=0; i<n; i++)
    {
        cin >> k;
        vec.push_back(k);
    }
    cout << "The number of occurences of '"<< vec[0] << "' in the array is: ";
    cout << count(vec.begin(), vec.end(), vec[0]) << endl;
    return 0;
}
```

Q01–b



```cpp
#include <iostream>
#include <cstdlib>

using namespace std;

int m, n;
int arr[1005][1005];
int ans[1005][1005];

void matrix_trans(int m, int n)
{
    int x;
    if(m>=n)
        x=m;
    else
        x=n;
    for(int i=1; i<x+1; i++)
    {
        for(int j=1; j<x+1; j++)
        {
            ans[i][j]=arr[j][i];
        }
    }
    for(int i=1; i<n+1; i++)
    {
        for(int j=1; j<m+1; j++)
        {
            cout << ans[i][j] << " ";
        }
        cout << endl;
    }
}

int main(void)
{
    cout << "Please decide the size of an 'm*n' matrix:" << endl;
    cout << "m = ";
    cin >> m;
    cout << "n = ";
    cin >> n;
    cout << "Please input the contain number of the matrix:" << endl;
    for(int i=1; i<m+1; i++)
    {
        for(int j=1; j<n+1; j++)
        {
            cin >> arr[i][j];
        }
    }
    matrix_trans(m, n);
    return 0;
}
```

Q02



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

$0^2+1^2+2^2+\dots+n^2 = \underbrace{1^2+2^2+3^2+\dots+n^2}_{n} \le \underbrace{n^2+n^2+\dots+n^2}_{n} = n \cdot n^2 = n^3$

$\because \forall n \in \mathbb{R} .\ n>0$

$0 \cdot n^3 \le \sum_{i=0}^{n} i^2 \le n^3$

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

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

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

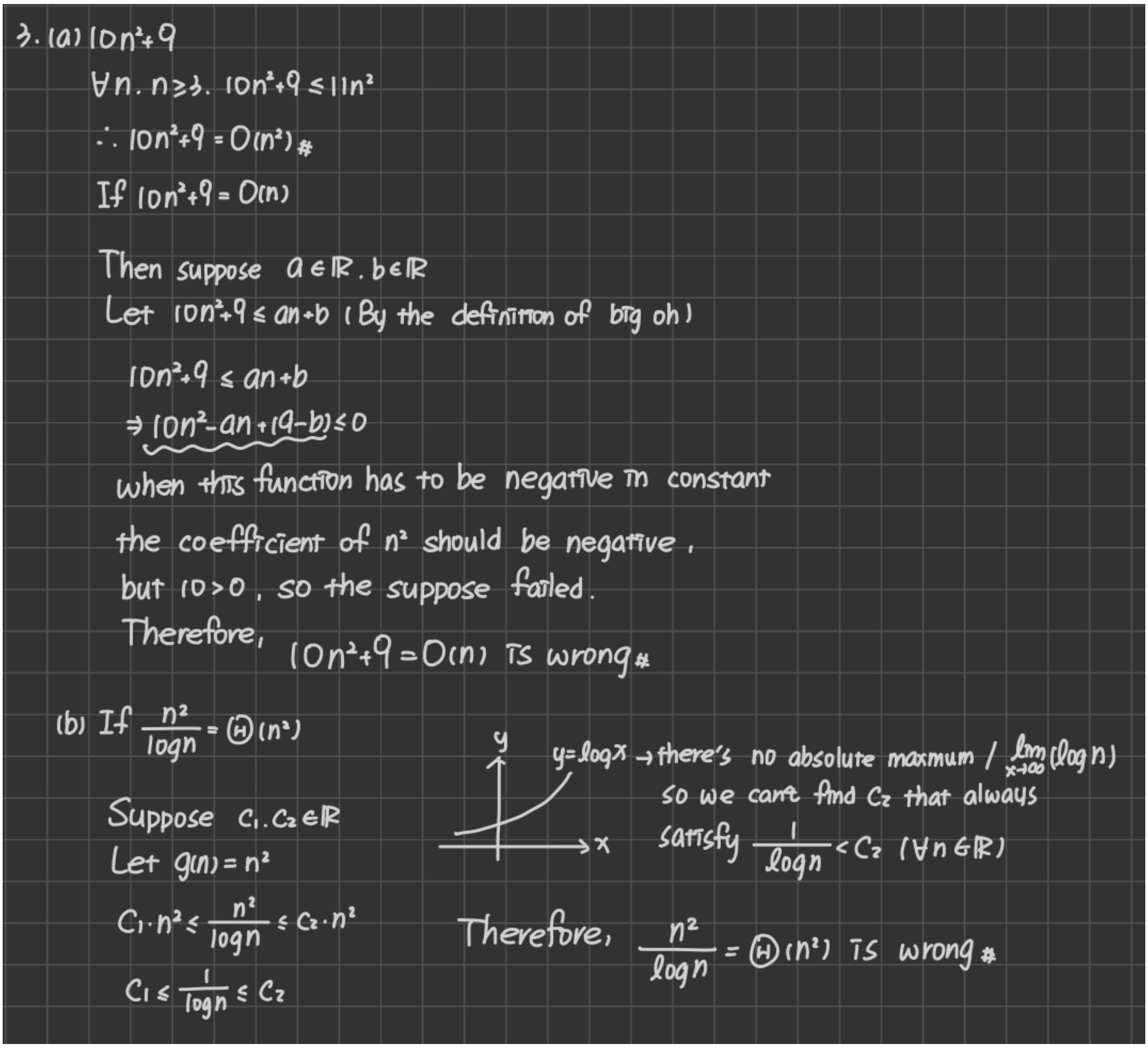
$\forall n.$ when $n>0$

$n! = \underbrace{1\times2\times\dots\times n}_{n} \le \underbrace{n\times n\times\dots\times n}_{n} = n^n$

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

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

Q03



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

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

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

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)

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

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

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 $g(n) = n^2$

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

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

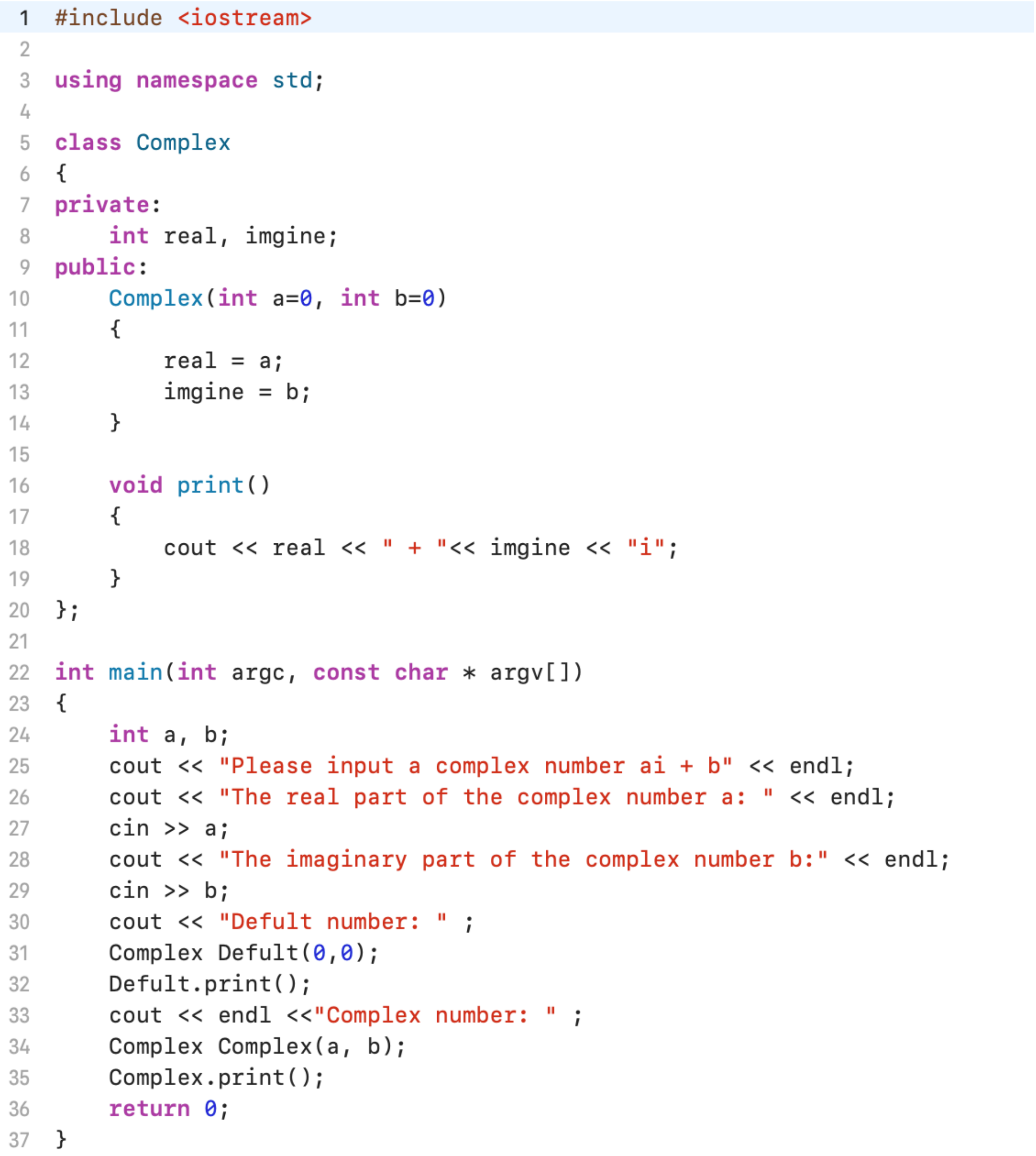
$y=\log x$ → there's no absolute maxmum / $\lim_{x\to\infty}(\log n)$

so we can't find c_2 that always

satisfy $\frac{1}{\log n} < c_2$ $(\forall n \in \mathbb{R})$

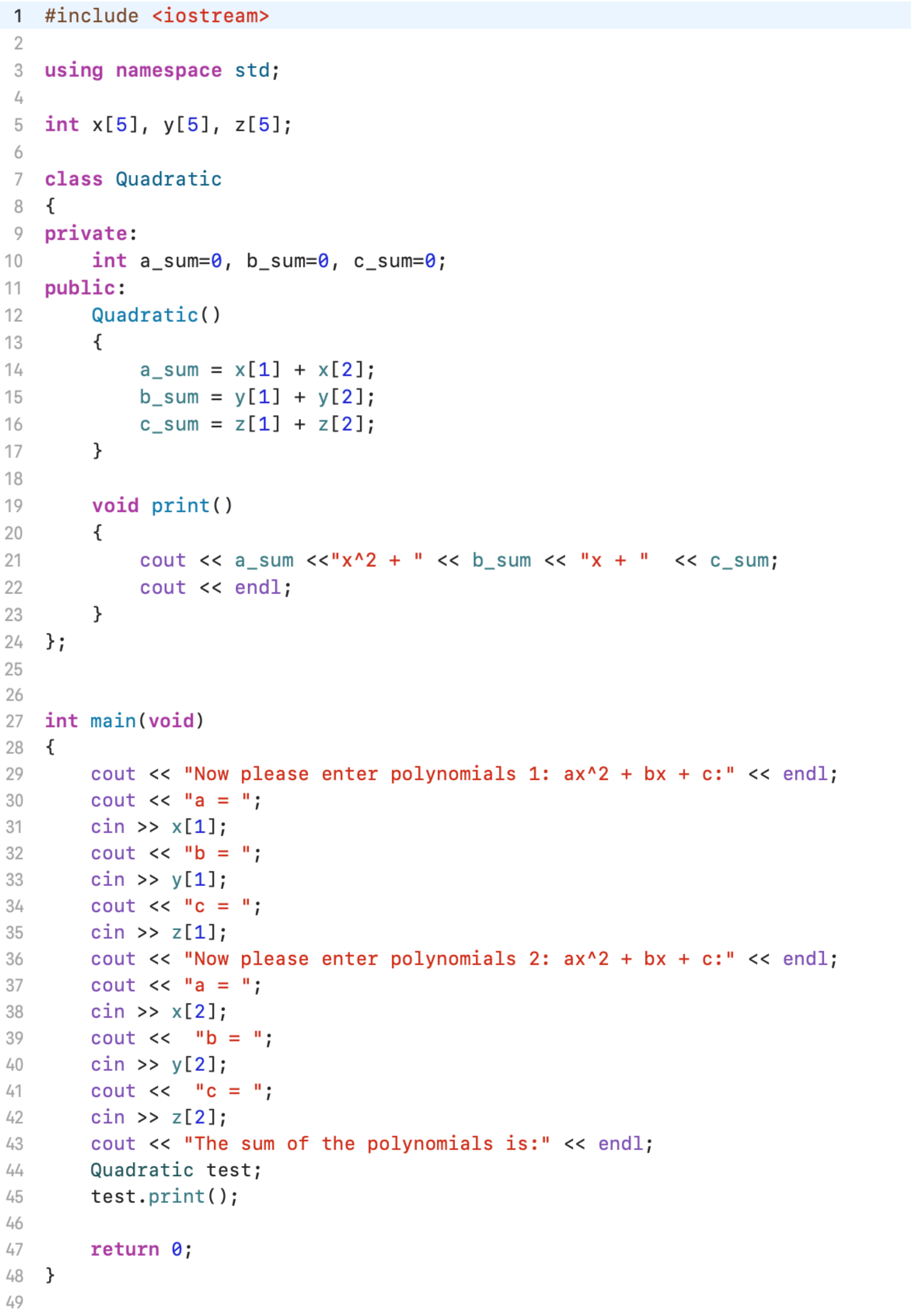
Therefore, $\frac{n^2}{\log n} = \Theta(n^2)$ is wrong #

Q04



```
#include <iostream>

using namespace std;

class Complex
{
private:
    int real, imgine;
public:
    Complex(int a=0, int b=0)
    {
        real = a;
        imgine = b;
    }

    void print()
    {
        cout << real << " + "<< imgine << "i";
    }
};

int main(int argc, const char * argv[])
{
    int a, b;
    cout << "Please input a complex number ai + b" << endl;
    cout << "The real part of the complex number a: " << endl;
    cin >> a;
    cout << "The imaginary part of the complex number b:" << endl;
    cin >> b;
    cout << "Defult number: " ;
    Complex Defult(0,0);
    Defult.print();
    cout << endl <<"Complex number: " ;
    Complex Complex(a, b);
    Complex.print();
    return 0;
}
```

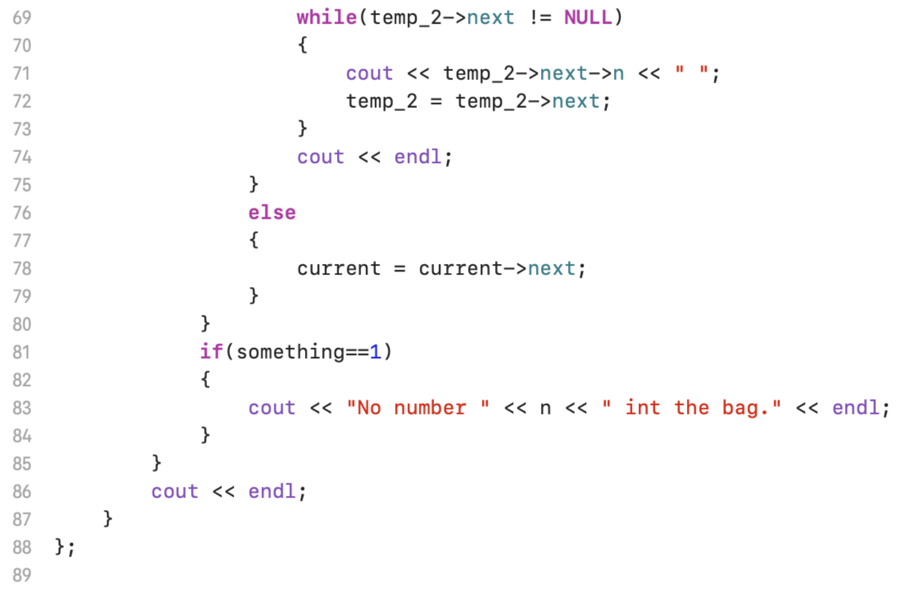
Q05



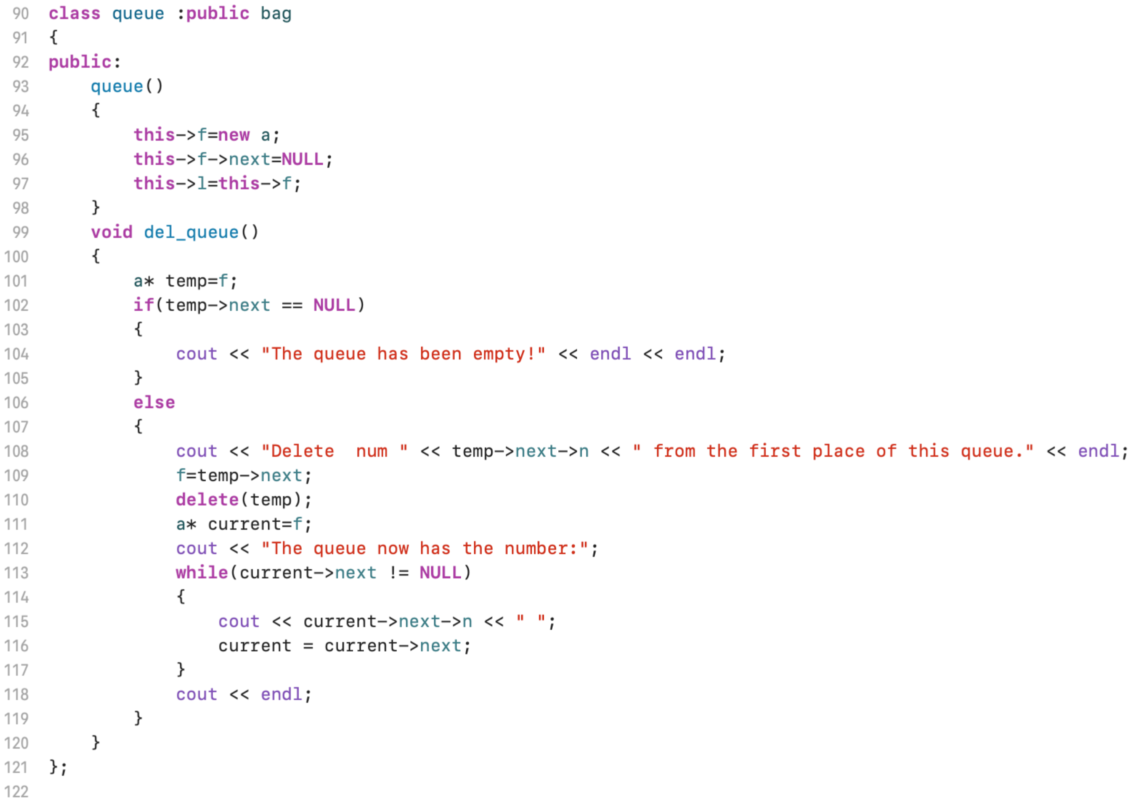
```cpp
#include <iostream>

using namespace std;

int x[5], y[5], z[5];

class Quadratic
{
private:
    int a_sum=0, b_sum=0, c_sum=0;
public:
    Quadratic()
    {
        a_sum = x[1] + x[2];
        b_sum = y[1] + y[2];
        c_sum = z[1] + z[2];
    }

    void print()
    {
        cout << a_sum <<"x^2 + " << b_sum << "x + "  << c_sum;
        cout << endl;
    }
};


int main(void)
{
    cout << "Now please enter polynomials 1: ax^2 + bx + c:" << endl;
    cout << "a = ";
    cin >> x[1];
    cout << "b = ";
    cin >> y[1];
    cout << "c = ";
    cin >> z[1];
    cout << "Now please enter polynomials 2: ax^2 + bx + c:" << endl;
    cout << "a = ";
    cin >> x[2];
    cout <<  "b = ";
    cin >> y[2];
    cout <<  "c = ";
    cin >> z[2];
    cout << "The sum of the polynomials is:" << endl;
    Quadratic test;
    test.print();

    return 0;
}

```

Q06

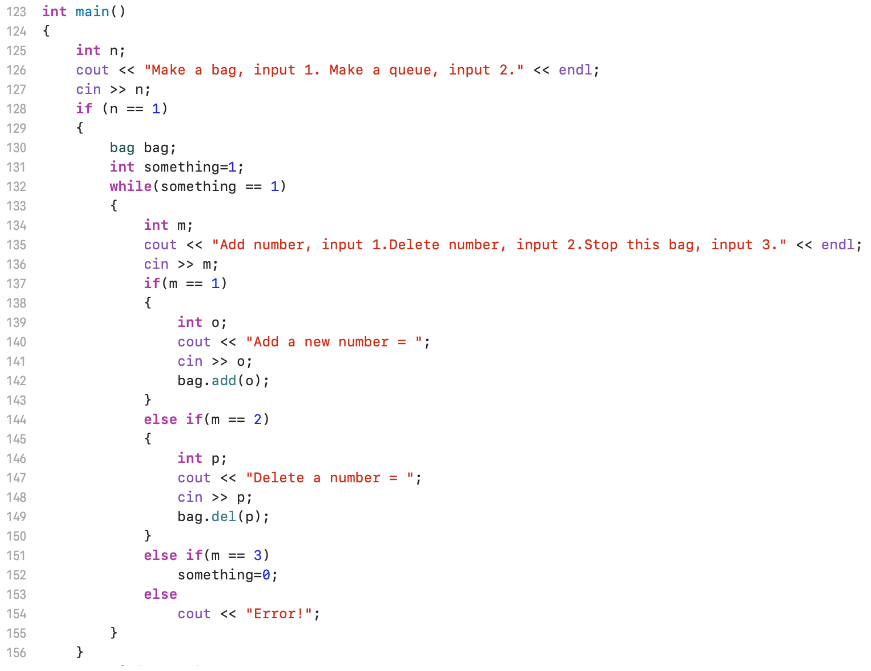
```cpp
#include<iostream>

using namespace std;

struct a
{
    int n;
    a* next;
};

class bag
{
public:
    a* f, * l;

    bag()
    {
        f=new a;
        f->next=NULL;
        l=f;
    }

    void add(int n)
    {
        a* temp=new a;
        temp->n=n;
        temp->next=NULL;
        l->next=temp;
        l=temp;
        a* current=f;
        cout << "Number:";
        while(current->next != NULL)
        {
            cout << current->next->n << " ";
            current=current->next;
        }
        cout << endl;
    }

    void initialize()
    {
        f=new a;
        f->next=NULL;
        l=f;
    }

    void del(int n)
    {
        int something=1;
        int time=0;
        a* current=f;
        if(current == NULL)
        {
            cout << n << " isn't in the bag." << endl;
        }
        else
        {
            while(current->next != NULL)
            {
                if(current->next->n == n)
                {
                    something=0;
                    a* temp = current->next;
                    current->next = temp->next;
                    delete(temp);
                    time++;
                    cout << "The bag had delete " << time << " time of the number " << n << ":";
                    a* temp_2=f;
```



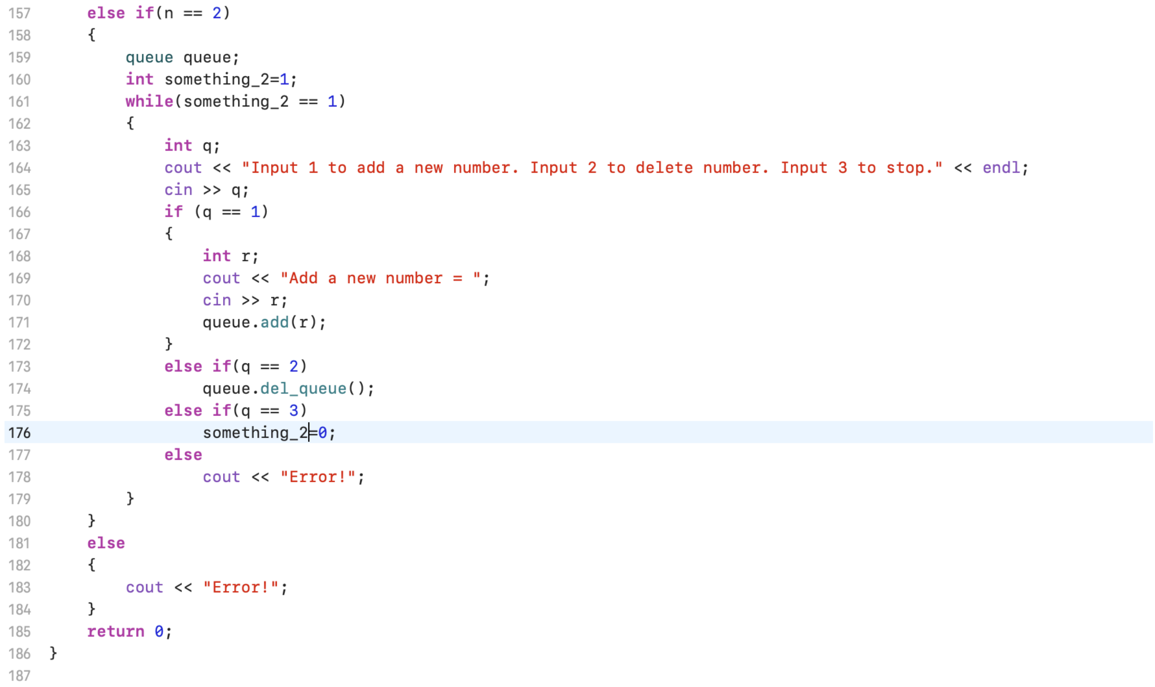
```
                    while(temp_2->next != NULL)
                    {
                        cout << temp_2->next->n << " ";
                        temp_2 = temp_2->next;
                    }
                    cout << endl;
                }
                else
                {
                    current = current->next;
                }
            }
            if(something==1)
            {
                cout << "No number " << n << " int the bag." << endl;
            }
        }
        cout << endl;
    }
};

```



```
class queue :public bag
{
public:
    queue()
    {
        this->f=new a;
        this->f->next=NULL;
        this->l=this->f;
    }
    void del_queue()
    {
        a* temp=f;
        if(temp->next == NULL)
        {
            cout << "The queue has been empty!" << endl << endl;
        }
        else
        {
            cout << "Delete  num " << temp->next->n << " from the first place of this queue." << endl;
            f=temp->next;
            delete(temp);
            a* current=f;
            cout << "The queue now has the number:";
            while(current->next != NULL)
            {
                cout << current->next->n << " ";
                current = current->next;
            }
            cout << endl;
        }
    }
};

```

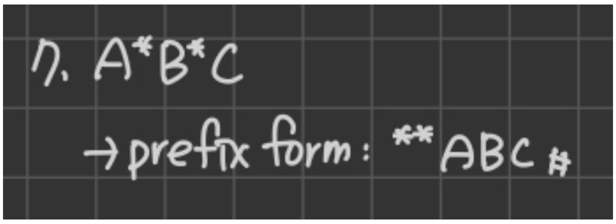


```
int main()
{
    int n;
    cout << "Make a bag, input 1. Make a queue, input 2." << endl;
    cin >> n;
    if (n == 1)
    {
        bag bag;
        int something=1;
        while(something == 1)
        {
            int m;
            cout << "Add number, input 1.Delete number, input 2.Stop this bag, input 3." << endl;
            cin >> m;
            if(m == 1)
            {
                int o;
                cout << "Add a new number = ";
                cin >> o;
                bag.add(o);
            }
            else if(m == 2)
            {
                int p;
                cout << "Delete a number = ";
                cin >> p;
                bag.del(p);
            }
            else if(m == 3)
                something=0;
            else
                cout << "Error!";
        }
    }
```



```cpp
    else if(n == 2)
    {
        queue queue;
        int something_2=1;
        while(something_2 == 1)
        {
            int q;
            cout << "Input 1 to add a new number. Input 2 to delete number. Input 3 to stop." << endl;
            cin >> q;
            if (q == 1)
            {
                int r;
                cout << "Add a new number = ";
                cin >> r;
                queue.add(r);
            }
            else if(q == 2)
                queue.del_queue();
            else if(q == 3)
                something_2=0;
            else
                cout << "Error!";
        }
    }
    else
    {
        cout << "Error!";
    }
    return 0;
}
```

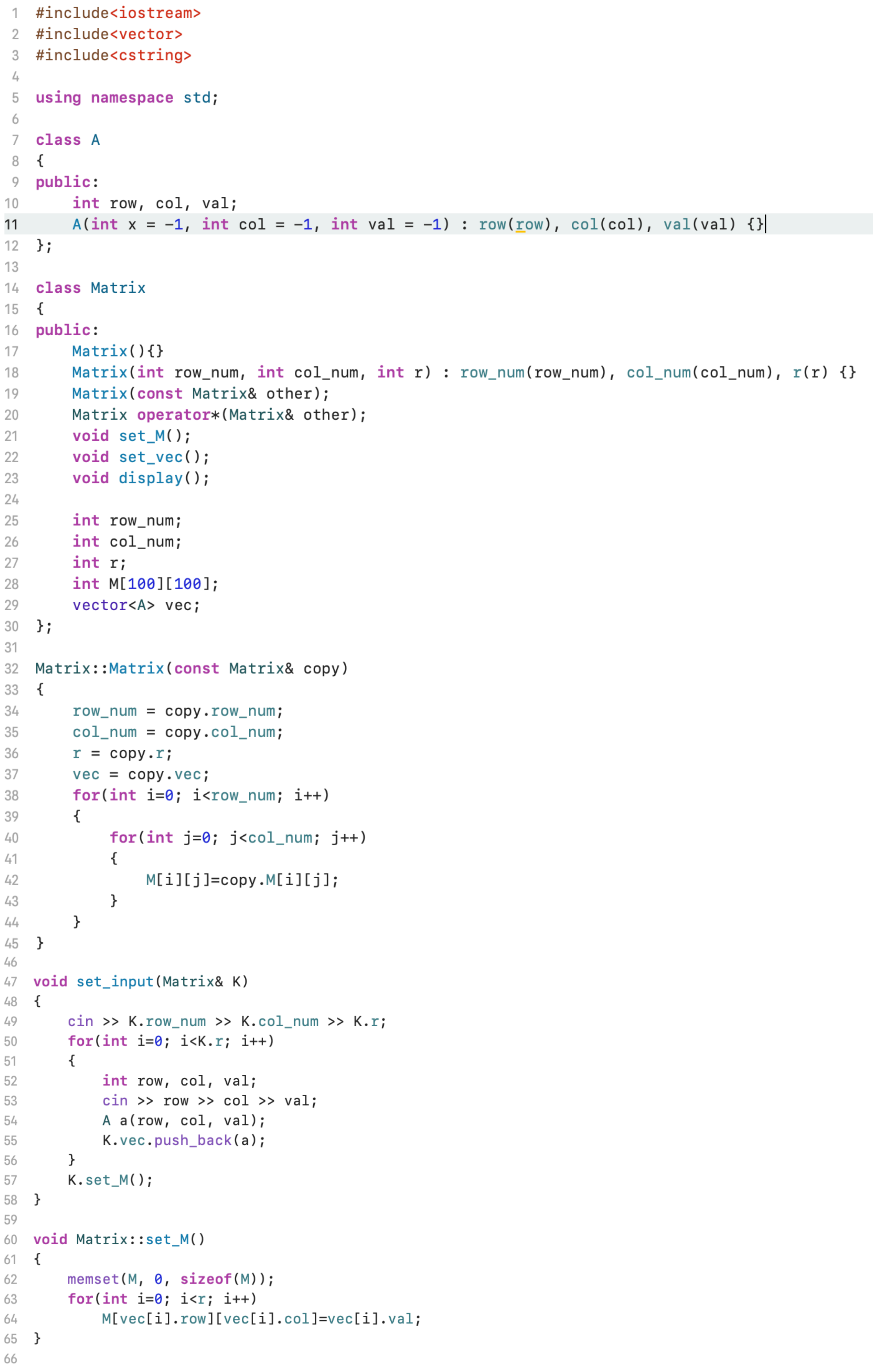
Q07



7. A*B*C

→ prefix form: **ABC #

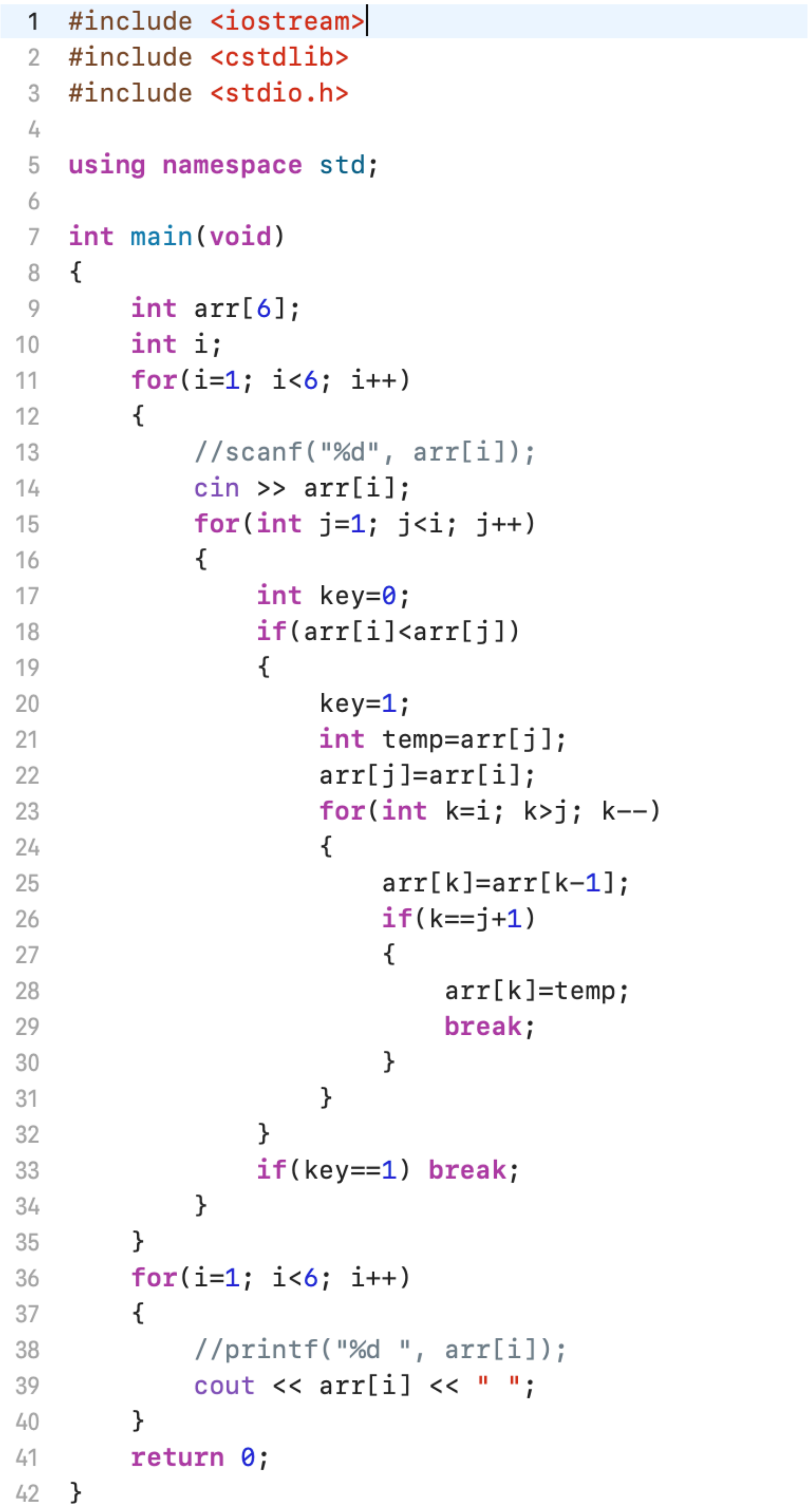
Q08



```cpp
#include<iostream>
#include<vector>
#include<cstring>

using namespace std;

class A
{
public:
    int row, col, val;
    A(int x = -1, int col = -1, int val = -1) : row(row), col(col), val(val) {}
};

class Matrix
{
public:
    Matrix(){}
    Matrix(int row_num, int col_num, int r) : row_num(row_num), col_num(col_num), r(r) {}
    Matrix(const Matrix& other);
    Matrix operator*(Matrix& other);
    void set_M();
    void set_vec();
    void display();

    int row_num;
    int col_num;
    int r;
    int M[100][100];
    vector<A> vec;
};

Matrix::Matrix(const Matrix& copy)
{
    row_num = copy.row_num;
    col_num = copy.col_num;
    r = copy.r;
    vec = copy.vec;
    for(int i=0; i<row_num; i++)
    {
        for(int j=0; j<col_num; j++)
        {
            M[i][j]=copy.M[i][j];
        }
    }
}

void set_input(Matrix& K)
{
    cin >> K.row_num >> K.col_num >> K.r;
    for(int i=0; i<K.r; i++)
    {
        int row, col, val;
        cin >> row >> col >> val;
        A a(row, col, val);
        K.vec.push_back(a);
    }
    K.set_M();
}

void Matrix::set_M()
{
    memset(M, 0, sizeof(M));
    for(int i=0; i<r; i++)
        M[vec[i].row][vec[i].col]=vec[i].val;
}
```

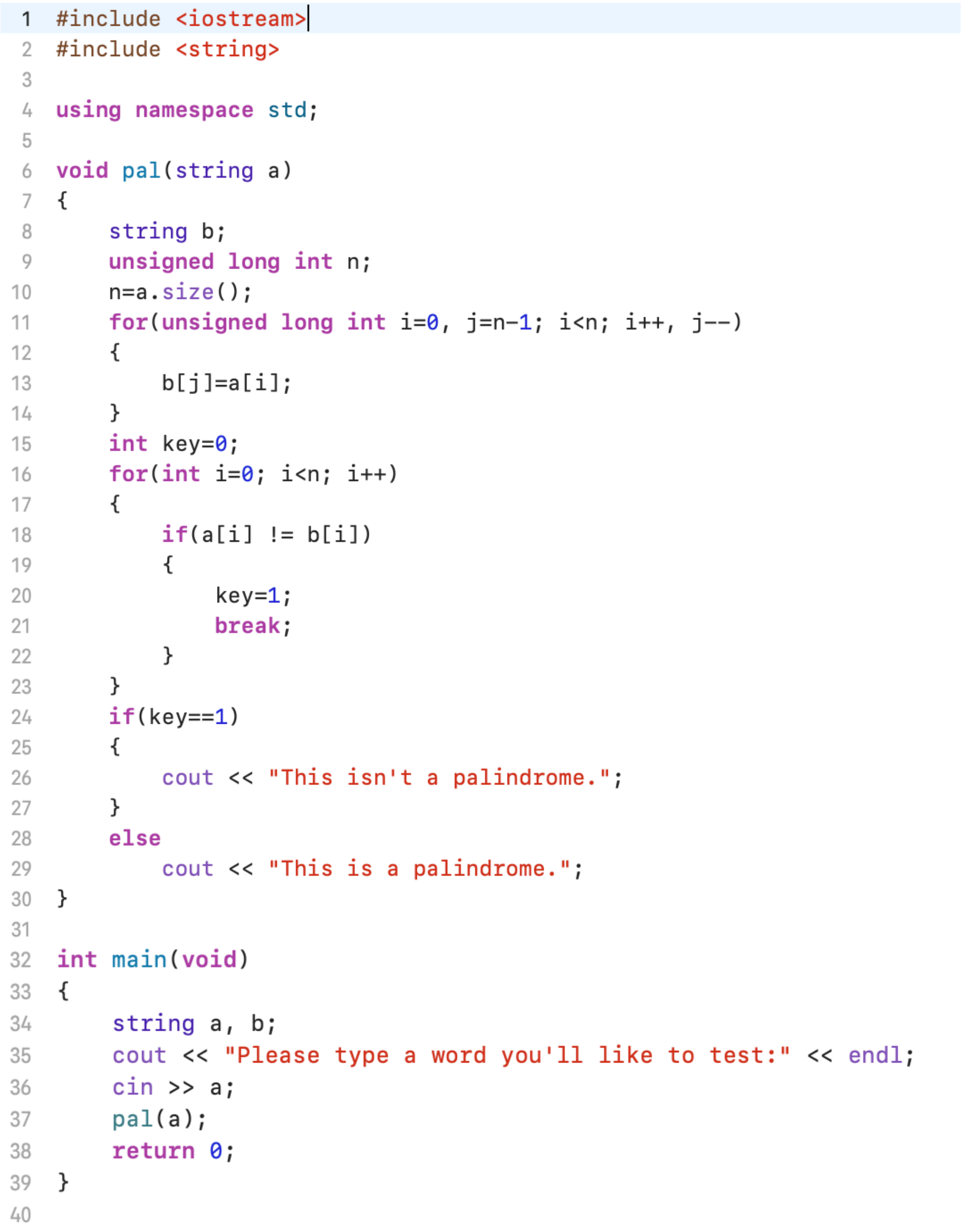
```cpp
void Matrix::set_vec()
{
    int count=0;
    for(int i=0; i<row_num; i++)
    {
        for(int j=0; j<col_num; j++)
        {
            if (M[i][j] != 0)
            {
                A element(i, j, M[i][j]);
                vec.push_back(element);
                count++;
            }
        }
    }
    r = count;
}

void Matrix::display()
{
    for(int i=0; i<row_num; i++)
    {
        for(int j=0; j<col_num; j++)
            cout << M[i][j] << " ";
        cout << endl;
    }
}

Matrix Matrix::operator*(Matrix& other)
{
    Matrix mul_f(row_num, other.col_num, 0);

    if(other.col_num*r <= row_num*other.r)
    {
        for(int i=0; i<r; i++)
        {
            for(int j=0; j<other.col_num; j++)
                mul_f.M[vec[i].row][j] += vec[i].val*other.M[vec[i].col][j];
        }
    }
    else
    {
        for(int i=0; i<other.r; i++)
        {
            for(int j=0; j<row_num; j++)
                mul_f.M[j][other.vec[i].col] += other.vec[i].val*M[j][other.vec[i].row];
        }
    }
    mul_f.set_vec();
    return mul_f;
}

int main()
{
    Matrix Sample;
    Matrix Sample_2;
    set_input(Sample);
    set_input(Sample_2);

    Matrix copy_SM=Sample;
    cout << "copy result: " << endl;
    copy_SM.display();
    return 0;
}
```

Q09



```cpp
#include <iostream>
#include <cstdlib>
#include <stdio.h>

using namespace std;

int main(void)
{
    int arr[6];
    int i;
    for(i=1; i<6; i++)
    {
        //scanf("%d", arr[i]);
        cin >> arr[i];
        for(int j=1; j<i; j++)
        {
            int key=0;
            if(arr[i]<arr[j])
            {
                key=1;
                int temp=arr[j];
                arr[j]=arr[i];
                for(int k=i; k>j; k--)
                {
                    arr[k]=arr[k-1];
                    if(k==j+1)
                    {
                        arr[k]=temp;
                        break;
                    }
                }
            }
            if(key==1) break;
        }
    }
    for(i=1; i<6; i++)
    {
        //printf("%d ", arr[i]);
        cout << arr[i] << " ";
    }
    return 0;
}
```

Q10



```cpp
#include <iostream>
#include <string>

using namespace std;

void pal(string a)
{
    string b;
    unsigned long int n;
    n=a.size();
    for(unsigned long int i=0, j=n-1; i<n; i++, j--)
    {
        b[j]=a[i];
    }
    int key=0;
    for(int i=0; i<n; i++)
    {
        if(a[i] != b[i])
        {
            key=1;
            break;
        }
    }
    if(key==1)
    {
        cout << "This isn't a palindrome.";
    }
    else
        cout << "This is a palindrome.";
}

int main(void)
{
    string a, b;
    cout << "Please type a word you'll like to test:" << endl;
    cin >> a;
    pal(a);
    return 0;
}

```