

Задача 1

Отсортировать столбцы матрицы методом вставки.

Если во входном файле input.txt записано

```
7
26 39 33 21 51 77 42
11 21 23 56 87 62 10
52 23 76 62 76 11 77
54 19 60 27 18 43 43
78 32 75 28 55 64 31
66 41 82 41 76 44 34
88 23 52 91 53 46 12
```

то в выходной файл output.txt должно быть записано

```
11 19 23 21 18 11 10
26 21 33 27 51 43 12
52 23 52 28 53 44 31
54 23 60 41 55 46 34
66 32 75 56 76 62 42
78 39 76 62 76 64 43
88 41 82 91 87 77 77
```

```
#include <fstream>
#include <iostream>
#include <vector>
using namespace std;
ifstream in("input.txt");
ofstream out("output.txt");

void Sort(vector<int> &a)
//сортирует вектор a методом вставки
{
    for (int i = 1; i < a.size(); i++) {
        int j = i;
        while (j > 0 && a[j] < a[j - 1]) {
            swap(a[j - 1], a[j]);
            j--;
        }
    }
}

int main()
{
    int n, x;
    in >> n;
    vector<vector<int>> mas(n);
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++) {
            in >> x;
            mas[i].push_back(x);
        }
}
```

```

vector<int> b;
for (int j = 0; j < n; j++) {
    for (int i = 0; i < n; i++)
    {
        b.push_back(mas[i][j]);
    }
    Sort(b);
    for (int i = 0; i < n; i++) {
        mas[i][j] = b[i];
    }
    b.clear();
}
for (int i = 0; i < mas.size(); i++) {
    for (int j = 0; j < mas[i].size(); j++)
        out << mas[i][j] << "\t";
    out << endl;
}
system("pause");
return 0;
}

```

Задача 2

Отсортировать диагонали матрицы, параллельные главной методом подсчета.

Если во входном файле input.txt записано

```

7
5 4 1 2 3 6 7
8 5 4 7 9 6 2
0 1 2 0 1 2 3
2 2 2 6 3 2 6
7 4 7 4 7 8 5
9 6 9 5 2 0 5
7 4 5 8 6 9 1

```

то в выходной файл output.txt должно быть записано

```

0    0    1    2    3    2    7
1    1    3    1    2    3    6
0    2    2    4    2    6    6
2    2    2    5    4    5    9
5    4    5    4    5    5    7
4    6    8    6    8    6    8
7    9    7    9    7    9    7

```

```

#include <fstream>
#include <iostream>
#include <vector>
using namespace std;
ifstream in("input.txt");
ofstream out("output.txt");

```

```

void Sort(vector<int> &a)
//сортирует вектор a методом подсчета
{
    int max = a[0], min = a[0];
    for (int i = 1; i < a.size(); i++) {
        if (a[i] > max)
            max = a[i];
        if (a[i] < min)
            min = a[i];
    }
    vector <int> c(max - min + 1, 0);

    for (int i = 0; i < a.size(); i++) {
        c[a[i] - min] ++;
    }
    a.clear();
    for (int i = 0 ; i <= max-min; i++){
        for (int j = 0; j < c[i]; j++)
            a.push_back(i+min);
    }
}

int main()
{
    int n, x;
    in >> n;
    vector<vector<int>> mas(n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            in >> x;
            mas[i].push_back(x);
        }
    }
    vector <int>b;
    //Диагонали параллельные главной и выше главной
    for (int k = 0; k<n; k++)
    {
        int i, j;
        for ( j = k, i=0; j < n ; j++, i++)
        {
            b.push_back(mas[i][j]);
        }
        Sort(b);
        for (j = k, i = 0; j < n; j++, i++)
        {
            mas[i][j] = b[i];
        }
        b.clear();
    }
}

```

```

// Диагонали параллельные главной и ниже главной
for (int k = 1; k < n; k++)
{
    int i, j;
    for (j = 0, i = k; i < n; j++, i++)
    {
        b.push_back(mas[i][j]);
    }
    Sort(b);
    for (j = 0, i = k; i < n; j++, i++)
    {
        mas[i][j] = b[j];
    }
    b.clear();
}
for (int i = 0; i < mas.size(); i++) {
    for (int j = 0; j < mas[i].size(); j++)
        out << mas[i][j] << "\t";
    out << endl;
}
system("pause");
return 0;
}

```

Задача 3

Отсортировать диагонали матрицы, параллельные побочной методом быстрой сортировки.

Если во входном файле input.txt записано

```

7
5 4 1 2 3 6 7
8 5 4 7 9 6 2
0 1 2 0 1 2 3
2 2 2 6 3 2 6
7 4 7 4 7 8 5
9 6 9 5 2 0 5
7 4 5 8 6 9 1

```

то в выходной файл output.txt должно быть записано

```

5      8      5      4      7      9      7
4      1      2      7      9      7      9
0      2      3      6      7      4      7
1      2      4      6      4      5      8
2      2      6      3      5      8      6
0      6      2      3      6      5      9
1      2      2      2      0      5      1

```

```

#include <fstream>
#include <iostream>
#include <vector>
using namespace std;
ifstream in("input.txt");

```

```

ofstream out("output.txt");

void Sort(vector<int> &a, int l, int r)
//сортирует вектор a быстрой сортировкой
{
    int i = l, j = r;
    int el = a[(r + l) / 2];
    while (i <= j) {
        while (a[i] < el)
            i++;
        while (a[j] > el)
            j--;
        if (i <= j) {
            swap(a[i], a[j]);
            i++;
            j--;
        }
    }
    if (l < j)
        Sort(a, l, j);
    if (i < r)
        Sort(a, i, r);
}

int main()
{
    int n, x;
    in >> n;
    vector<vector<int>> mas(n);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            in >> x;
            mas[i].push_back(x);
        }
    }
    vector<int> b;
    //диагонали параллельные побочной и выше главной
    for (int k = 1; k < n; k++)
    {
        int i, j;
        for (j = 0, i = k; i >= 0; j++, i--)
        {
            b.push_back(mas[i][j]);
        }
        Sort(b, 0, b.size()-1);
        for (j = 0, i = k; i >= 0; j++, i--)
        {
            mas[i][j] = b[j];
        }
        b.clear();
    }
}

```

```

// Диагонали параллельные побочной и ниже главной
for (int k = 1; k < n; k++)
{
    int i, j, p;
    for (j = k, i = n-1; j < n; j++, i--)
    {
        b.push_back(mas[i][j]);
    }
    Sort(b, 0, b.size() - 1);
    for (j = k, i = n - 1, p = 0; j < n; j++, i--, p++)
    {
        mas[i][j] = b[p];
    }
    b.clear();
}
for (int i = 0; i < mas.size(); i++) {
    for (int j = 0; j < mas[i].size(); j++)
        out << mas[i][j] << "\t";
    out << endl;
}
return 0;
}

```