

```

st2=1;
for (j=1; j<=n+1; j++)
    st2+=a[1][j];
cout << " " << st2 << endl;
//cout << " ";
//for (j=0; j<=n; j++)
//    cout <<a[1][j];
//cout << endl;

return 0;
}

```

Галина Момчева

Задача D3. РЕДИЦИ

Дадени са n редици, номерирани с числата от 1 до n . Всяка от тях съдържа по n цели числа, подредени в нарастващ ред. Дадено е още и цяло число x .

Напишете програма **d3**, която размества числата в редиците по следния алгоритъм:

```

=====
Направете  $n$  стъпки, за  $i=1, 2, \dots, n$ .
На  $i$ -тата стъпка разменете стойностите на  $x$  и на най-малкото число от  $i$ -тата редица, което е по-голямо или равно на  $x$ . Ако всички числа от  $i$ -тата редица са по-малки от  $x$ , разменете числото  $x$  с най-голямото число от редицата.
=====

```

От първия ред на стандартния вход се въвежда число n ($0 < n < 200$), задаващо броя на редиците.

От втория ред се въвежда числото x . От следващите n реда се въвеждат n растящи редици с по n елемента всяка. Стойностите на елементите в редиците не надвишават 25000.

На стандартния изход да се изведат числата, с които се разменя x при изпълнението на алгоритъма. Числата да бъдат на един ред, разделени с по един интервал.

ПРИМЕР

Вход

```

3
2
1 3 5
7 8 9
2 3 4

```

Изход

```

3 7 4

```

РЕШЕНИЕ

Идеята на задачата е от теорията на диаграмите на Юнг, макар че ограниченията към редиците там са по-строги и когато се срещне елемент, който не може да бъде замесен се допълва реда с него.

```
#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int n, i=0, j, x;
    int a[200];
    int b[200];
    cin >> n;
    cin >> x;
    while (i<n)
    {
        for (j=0; j<n; j++)
            cin >> a[j];
        j=0;
        while (j<n && a[j]<x)
            j++;
        if (j>=n) {cout << a[j-1]; swap(a[j-1], x); }
        else if (a[j]>=x) {cout << a[j]; swap(a[j], x);}
        if (i<n-1) cout << ' ';
        i++;
    }

    cout << endl;
    return 0;
}
```

Галина Момчева