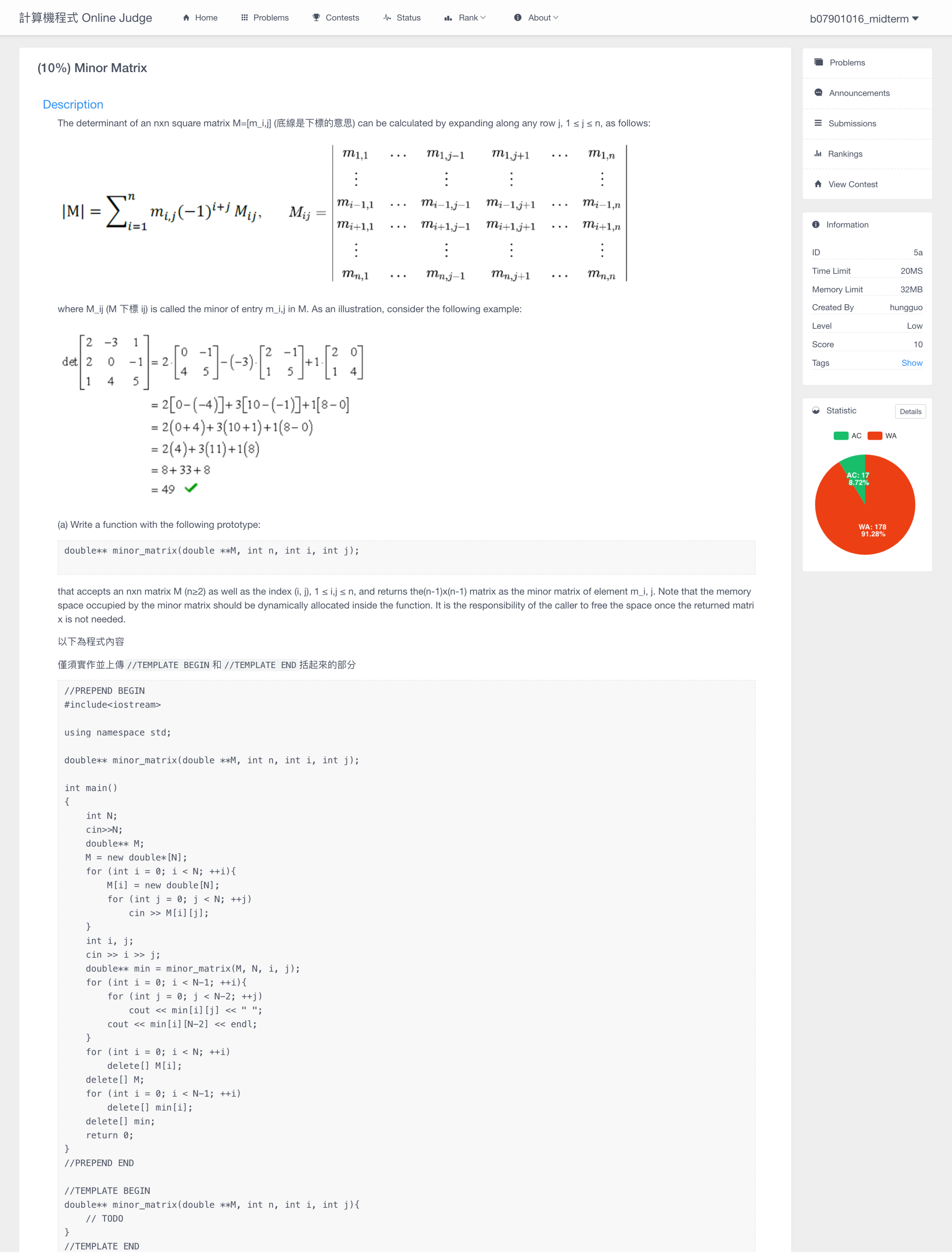


(10%) Minor Matrix

Description

The determinant of an nxn square matrix M=[m_i,j] (底線是下標的意思) can be calculated by expanding along any row j, 1 ≤ j ≤ n, as follows:

$$|M| = \sum_{i=1}^{n} m_{i,j}(-1)^{i+j} M_{ij}, \qquad M_{ij} = \begin{vmatrix} m_{1,1} & \cdots & m_{1,j-1} & m_{1,j+1} & \cdots & m_{1,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ m_{i-1,1} & \cdots & m_{i-1,j-1} & m_{i-1,j+1} & \cdots & m_{i-1,n} \\ m_{i+1,1} & \cdots & m_{i+1,j-1} & m_{i+1,j+1} & \cdots & m_{i+1,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ m_{n,1} & \cdots & m_{n,j-1} & m_{n,j+1} & \cdots & m_{n,n} \end{vmatrix}$$

where M_ij (M 下標 ij) is called the minor of entry m_i,j in M. As an illustration, consider the following example:

$$\begin{aligned} \det\begin{bmatrix} 2 & -3 & 1 \\ 2 & 0 & -1 \\ 1 & 4 & 5 \end{bmatrix} &= 2\cdot\begin{bmatrix} 0 & -1 \\ 4 & 5 \end{bmatrix} - (-3)\cdot\begin{bmatrix} 2 & -1 \\ 1 & 5 \end{bmatrix} + 1\cdot\begin{bmatrix} 2 & 0 \\ 1 & 4 \end{bmatrix} \\ &= 2[0-(-4)] + 3[10-(-1)] + 1[8-0] \\ &= 2(0+4) + 3(10+1) + 1(8-0) \\ &= 2(4) + 3(11) + 1(8) \\ &= 8+33+8 \\ &= 49 \ \checkmark \end{aligned}$$

(a) Write a function with the following prototype:

```
double** minor_matrix(double **M, int n, int i, int j);
```

that accepts an nxn matrix M (n≥2) as well as the index (i, j), 1 ≤ i,j ≤ n, and returns the(n-1)x(n-1) matrix as the minor matrix of element m_i, j. Note that the memory space occupied by the minor matrix should be dynamically allocated inside the function. It is the responsibility of the caller to free the space once the returned matrix is not needed.

以下為程式內容

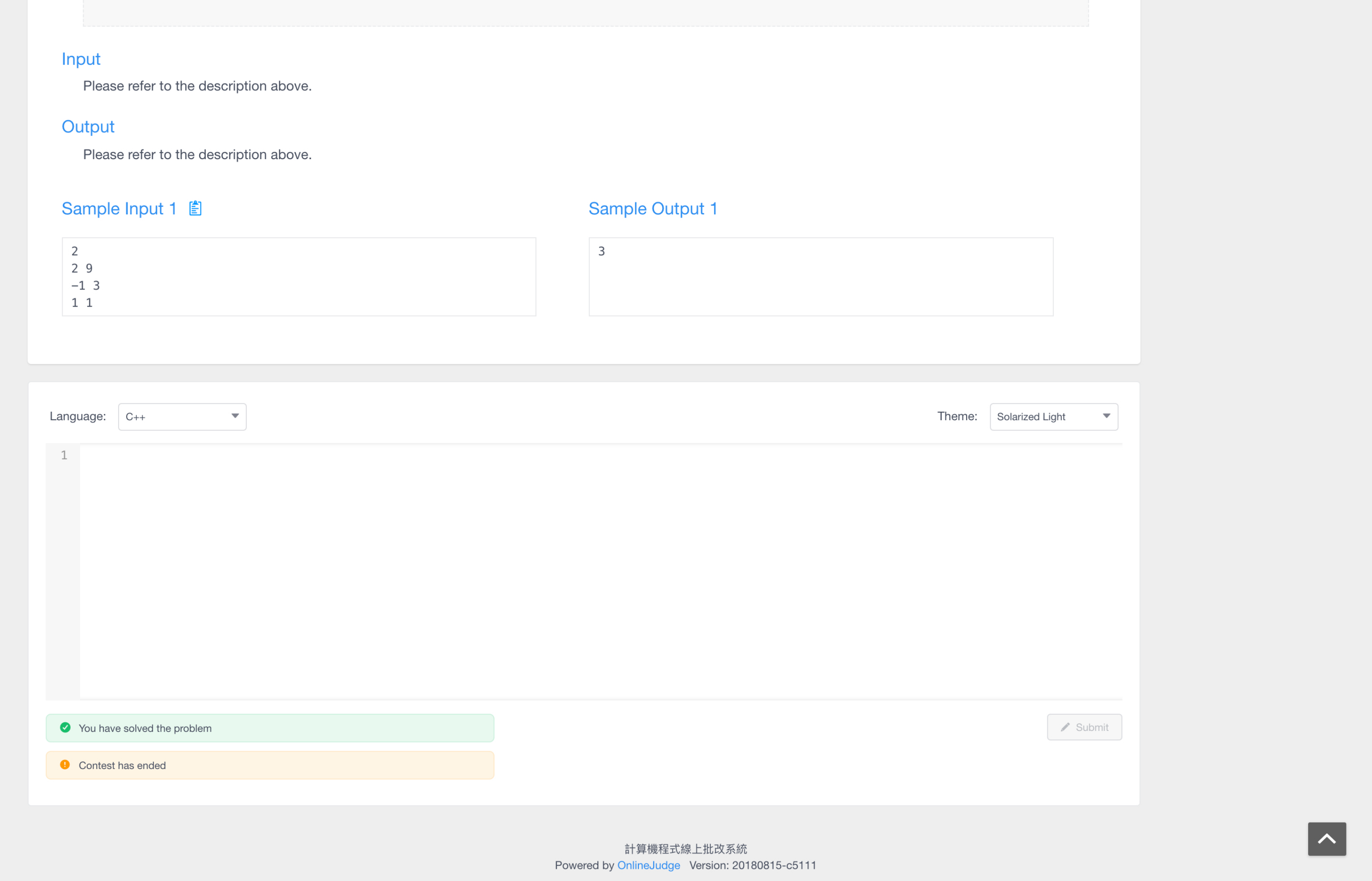
僅須實作並上傳 `//TEMPLATE BEGIN` 和 `//TEMPLATE END` 括起來的部分

```
//PREPEND BEGIN
#include<iostream>

using namespace std;

double** minor_matrix(double **M, int n, int i, int j);

int main()
{
    int N;
    cin>>N;
    double** M;
    M = new double*[N];
    for (int i = 0; i < N; ++i){
        M[i] = new double[N];
        for (int j = 0; j < N; ++j)
            cin >> M[i][j];
    }
    int i, j;
    cin >> i >> j;
    double** min = minor_matrix(M, N, i, j);
    for (int i = 0; i < N-1; ++i){
        for (int j = 0; j < N-2; ++j)
            cout << min[i][j] << " ";
        cout << min[i][N-2] << endl;
    }
    for (int i = 0; i < N; ++i)
        delete[] M[i];
    delete[] M;
    for (int i = 0; i < N-1; ++i)
        delete[] min[i];
    delete[] min;
    return 0;
}
//PREPEND END

//TEMPLATE BEGIN
double** minor_matrix(double **M, int n, int i, int j){
    // TODO
}
//TEMPLATE END
```

Information

ID	5a
Time Limit	20MS
Memory Limit	32MB
Created By	hungguo
Level	Low
Score	10
Tags	Show

Statistic



Input

Please refer to the description above.

Output

Please refer to the description above.

Sample Input 1

```
2
2 9
-1 3
1 1
```

Sample Output 1

```
3
```

Language: C++

Theme: Solarized Light

You have solved the problem

Contest has ended

計算機程式線上批改系統
Powered by OnlineJudge Version: 20180815-c5111