

# DP 测试题

(请选手务必仔细阅读本页内容)

## 一、题目概况

|         |                  |             |          |               |
|---------|------------------|-------------|----------|---------------|
| 中文题目名称  | 矩阵               | 产品排序        | 波动序列     | 逆序对           |
| 英文题目名称  | matrix           | product     | wave     | inversion     |
| 可执行文件名  | matrix           | product     | wave     | inversion     |
| 输入文件名   | matrix.in        | product.in  | wave.in  | inversion.in  |
| 输出文件名   | matrix.out       | product.out | wave.out | inversion.out |
| 每个测试点时限 | 2 秒              | 1 秒         | 1 秒      | 1 秒           |
| 测试点数目   | 20               | 20          | 10       | 10            |
| 每个测试点分值 | 5                | 5           | 10       | 10            |
| 结果比较方式  | 逐行比较(忽略多余空格和制表符) |             |          | 实数比较          |
| 附加样例文件  | 无                | 无           | 无        | 无             |
| 题目类型    | 传统               | 传统          | 传统       | 传统            |

## 二、提交源代码文件名

|            |            |             |          |               |
|------------|------------|-------------|----------|---------------|
| 对于pascal语言 | matrix.pas | product.pas | wave.pas | inversion.pas |
| 对于C语言      | matrix.c   | product.c   | wave.c   | inversion.c   |
| 对于C++语言    | matrix.cpp | product.cpp | wave.cpp | inversion.cpp |

## 三、编译命令 (不包含任何优化开关)

|              |                                 |                                   |                             |                                       |
|--------------|---------------------------------|-----------------------------------|-----------------------------|---------------------------------------|
| 对于 pascal 语言 | fpc matrix.pas                  | fpc product.pas                   | fpc wave.pas                | fpc inversion.pas                     |
| 对于 C 语言      | gcc -o matrix<br>matrix.c -lm   | gcc -o product<br>product.c -lm   | gcc -o wave<br>wave.c -lm   | gcc -o inversion<br>inversion.c -lm   |
| 对于C++语言      | g++ -o matrix<br>matrix.cpp -lm | g++ -o product<br>product.cpp -lm | g++ -o wave<br>wave.cpp -lm | g++ -o inversion<br>inversion.cpp -lm |

## 四、允许内存限制

|      |      |      |      |      |
|------|------|------|------|------|
| 内存上限 | 128M | 128M | 128M | 128M |
|------|------|------|------|------|

## 五、注意事项

- 1、每位选手提交一个以自己编号命名的文件夹，在该文件夹下放4个源程序，然后再在该文件夹下建立四个子目录，名称分别为：matrix、product、wave、inversion，再把源代码放入对应的子目录中。
- 2、文件夹名、文件名(程序名和输入输出文件名)必须使用英文小写。
- 3、C/C++中函数main()的返回值类型必须是int, 程序正常结束时的返回值必须是0。
- 4、统一评测时采用的机器配置为:windows下lemon评测和全国评测系统下评测。
- 5、最终测试时，所有编译命令均不打开任何优化开关。
- 6、请尽力优化，会收获更多的部分得分。

# 1. 矩阵

(`matrix.pas/c/cpp`)

【问题描述】

有一个  $n \times m$  的矩阵，你从左上角走到右下角，只能向下和向右走。  
每个点上有一个重量  $v_{ij}$  价值  $w_{ij}$  的物品，你有一个容量为  $S$  的背包，经过一个点你可以将此点的物品放入背包，求最大能得到的价值。

【输入】

输入文件 `matrix.in`。  
第一行三个数  $n, m, S$ 。  
下面  $n$  行，每行  $m$  个数，第  $i + 1$  行第  $j$  个数表示  $v_{ij}$ 。  
下面  $n$  行，每行  $m$  个数，第  $i + n + 1$  行第  $j$  个数表示  $w_{ij}$ 。

【输出】

输出文件 `matrix.out`。  
一行一个数表示最大的价值。

【输入输出样例】

| <code>matrix.in</code>   | <code>matrix.out</code> |
|--|-------------------------|
| 3 4 5<br>1 2 1 1<br>2 3 1 2<br>3 2 2 2<br>2 3 4 2<br>1 4 5 1<br>10 1 2 1 | 14                      |

【数据说明】

30%: $n \times m \leq 50, S \leq 100$   
60%: $n, m \leq 100, S \leq 200$   
100%: $n, m \leq 400, 0 \leq v_{ij}, w_{ij} \leq 1000, S \leq 400$

## 2. 产品排序

(product.pas/c/cpp)

### 【问题描述】

你是一个公司的员工，你会按时间顺序受到一些产品的订单，你需要用一个栈来改变这些订单的顺序（每个产品都必须入栈和出栈一次）。

按初始顺序，每次可以将一个产品入栈，或将栈顶产品弹至现在的序列末尾。每个产品有一个制作时间 $t_i$ 和单位时间惩罚值 $d_i$ ，总的惩罚值为 $\sum_{i=1}^n (s_i \times d_i)$ ，其中 $s_i$ 为第 $i$ 个产品的完成时间，你需要最小化总的惩罚值。

### 【输入】

输入文件 product.in。

第一行一个数  $n$ ，表示产品个数。

接下来  $n$  行，每行两个数表示 $t_i, d_i$ 。

### 【输出】

输出文件 product.out。

一行一个数表示最小的总惩罚值。

### 【输入输出样例】

| product.in | product.out |
|------------|-------------|
| 4          | 40          |
| 1 4        |             |
| 3 2        |             |
| 5 2        |             |
| 2 1        |             |

### 【样例解释】

| 操作步骤        | 排序后的序列(数字表示产品编号) |
|-------------|------------------|
| 1. 将第一个产品入栈 |                  |
| 2. 弹出栈顶元素   | 1                |
| 3. 将第二个产品入栈 | 1                |
| 4. 弹出栈顶元素   | 1 2              |
| 5. 将第三个产品入栈 | 1 2              |
| 6. 将第四个产品入栈 | 1 2              |
| 7. 弹出栈顶元素   | 1 2 4            |
| 8. 弹出栈顶元素   | 1 2 4 3          |

总惩罚值为  $1 \times 4 + (1+3) \times 2 + (1+3+2) \times 1 + (1+3+2+5) \times 2 = 40$

### 【数据说明】

30%: $n \leq 15$

50%: $n \leq 100$

100%: $n \leq 200, t_i, d_i \leq 1000$

### 3. 波动序列

(wave.pas/c/cpp)

**【问题描述】**

一个长度为  $n$  的序列  $\{a_i\}$  的波动值定义为  $\sum_{i=1}^{n-1} |a_{i-1} - a_i|$ 。  
你需要求出波动值为  $m$  的  $1 - n$  的排列个数模  $10^9 + 7$  的余数。

**【输入】**

输入文件 wave.in。  
第一行两个数  $n, m$ 。

**【输出】**

输出文件 wave.out。  
一行一个数表示答案。

**【输入输出样例】**

| wave.in | wave.out |
|---------|----------|
| 4 5     | 12       |

**【数据说明】**

30%:  $n \leq 10, m \leq 40$   
100%:  $n \leq 50$

## 4. 逆序对

(`inversion.pas/c/cpp`)

**【问题描述】**

有一个  $1 - n$  的排列，你会依次进行  $m$  次操作，第  $i$  次操作表示为  $(x_i, y_i)$ ，交换以这两个值为下标的元素，每次操作有一半的概率成功，你需要求出最后序列的逆序对的期望个数。

**【输入】**

输入文件 `inversion.in`。  
第一行两个数  $n, m$ 。  
第二行  $n$  个数表示初始的排列。  
接下来  $m$  行，每行两个数表示  $x_i, y_i$ 。

**【输出】**

输出文件 `inversion.out`。  
一个实数表示答案，四舍五入保留到小数点后 8 位，要求绝对误差不超过  $10^{-6}$ 。  
(评测时开启实数比较模式)

**【输入输出样例】**

| <code>inversion.in</code>           | <code>inversion.out</code> |
|-------------------------------------|----------------------------|
| 4 3<br>1 3 2 4<br>1 2<br>2 3<br>1 4 | 3.00000000                 |

**【数据说明】**

30%:  $n \leq 10, m \leq 20$   
100%:  $n \leq 1000, m \leq 1000$