

NOI98 第二试试题

围巾裁剪

裁缝有一块非常珍贵的丝绸围巾。可惜的是，围巾的某些部分已经被蛀虫给咬坏了。裁缝当然不愿意就这么把围巾给丢了，于是，他想把围巾给裁成两块小围巾送给他的两个女儿。自然，两块小围巾的面积之和越大越好。

这块围巾是一个正三角形，三条边被均匀地分成了 N 段，即这个正三角形被均匀地分成了 N^2 个单元，每个单元是一个面积为 1 的正三角形。图一所示为一个 $N=5$ 的围巾，图中带阴影的单元表示被蛀虫咬坏的部分。从上往下看，围巾被分成了 N 行，第一行有 1 个单元，第二行有 3 个单元，其中有 2 个是形如 Δ 的，有 1 个是形如 ∇ 的（这两种三角形我们认为是形状相同的）。第三行有 5 个，其中有 3 个是形如 Δ 的，有 2 个是形如 ∇ 的……。用坐标 (X,Y) 给每个单元定位，第一行的单元的坐标为 $(1,1)$ ；第二行从左到右的三个单元的坐标依次为 $(2,1)$ 、 $(2,2)$ 、 $(2,3)$ ；……。

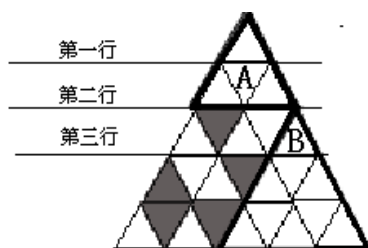


图 一

围巾的剪裁条件如下：

1. 裁成的两块小围巾形状与原来的大围巾完全相同，都是正三角形。
2. 每一块小围巾里都不存在被蛀虫咬坏的部分。
3. 裁剪时必须沿着单元的边界裁剪。
4. 要求两块小围巾的面积的和最大。

图一中，最优的裁剪方法已经用粗线画了出来，面积和为 $4+9=13$ 。

现在需要你编一个程序来帮助裁缝解决这个问题。

输入

输入文件第一行为一个整数 N ($1 \leq N \leq 100$)，表示这块围巾总共被分成了 N^2 个单元。第二行为一个整数 M ($0 \leq M \leq N^2-2$)，表示这块围巾共有 M 个单元被蛀虫咬坏了。接下的 M 行，每一行有两个正整数 X 和 Y ，为这 M 个被蛀虫咬坏的单元的坐标。

输入文件中同一行相邻两项之间用一个或多个空格隔开。

输出

输出文件仅含一个整数，为你所找到的裁出两块小围巾面积总和的最大值。

样例输入

```
5
5
3 2
4 1
4 4
5 4
5 2
```

样例输出

```
13
```

SERNET 模拟

计算机网络是现代科技发展的热点，传输性能是计算机网络的主要性能指标。SERKOI 网络开发小组设计了一种称为 SERNET 的网络，并希望开发一个模拟软件来模拟该网络的数据传输情况，进而计算出网络的传输性能。

SERNET 网络由服务器及连接它们的网络传输线路组成，服务器用服务器地址予以标识，网络传输线路为双向传输线路。网络传输过程中将各种待传输数据分割为若干个大小相同的数据包，以数据包为单位进行传输。数据包在传输线路上传输时需要一定的传输时间，不同的传输线路的传输时间不同。服务器处理数据的时间较之于传输时间很小，可忽略不计。每一个数据包中除了包括具体的数据信息外，还含有如下标识信息：

- 数据包编号

- 数据包源服务器地址

- 数据包目的服务器地址

网络传输的功能就是将一个个数据包从源服务器传输到目的服务器。对于每一个数据包，具体的网络传输方案为：

源服务器将待发送的数据包一律复制若干份并向与之相连的所有服务器发送该数据包。服务器接收到一个数据包后，如果该数据包符合下面任何一个条件：

- 数据包的源服务器地址与本服务器地址相同

- 数据包的目的服务器地址与本服务器地址相同

- 本服务器已转发过与该数据包编号相同的数据包

则接收该数据包；否则，服务器将其复制若干份并向与它相连的所有服务器转发该数据包。这里，两台服务器“相连”的含义指它们之间有网络传输线路直接相连。

现在需要你编一个程序来模拟 SERNET 网络中的数据包传输情况。

输入

输入文件的第一行为一个正整数 N ($N < 100$)，表示 SERNET 中服务器的数

目。第二行有 N 个互不相等的不超过 100 的正整数，表示每个服务器的地址。第三行有一个正整数 M，表示 SERNET 中传输线路的数目。接下来的 M 行每行用三个正整数表示一条传输线路连接的两台服务器的地址以及该传输线路的传输时间。线路传输时间为不超过 100 的正整数。接下来的一行为一个正整数 K ($K < 10000$)，表示 SERNET 中数据包的数目。以下的 K 行每行表示一个数据包的信息，格式为：

数据包编号 起始发送时间 源服务器地址 目的服务器地址
其中数据包编号为互不相同的小于 100000 的正整数。
输入文件的最后一行为一个正整数 T ($T < 10000$)，T 为输出时刻。
输入文件中同一行相邻两项之间用一个或多个空格隔开。

输出

输出文件的第一行为一个整数 P，表示 T 时刻后还在网络中传输的数据包数目（编号相同的数据包为同一数据包）。

约定

本题中所有时间量的单位均相同；
每一条传输线路上在同一时刻能传输任意多个数据包。

样例输入

```
4
57 42 10 93
4
57 42 6
42 93 5
42 10 2
10 93 10
2
433 10 57 10
5678 11 42 93
23
```

样例输出

```
1
```

并行计算

运算器（ALU）是计算机中的重要部件，它的功能是进行数学运算。图一是运算器的工作简图，运算器的一次运算操作过程为：运算器在控制器的控制下，从指定的存储器（MEMORY）存储单元中读出待运算的两个源操作数 A 和 B，经过一定时间的计算后得到运算结果 C，并将它写入指定的存储器存储单元中。

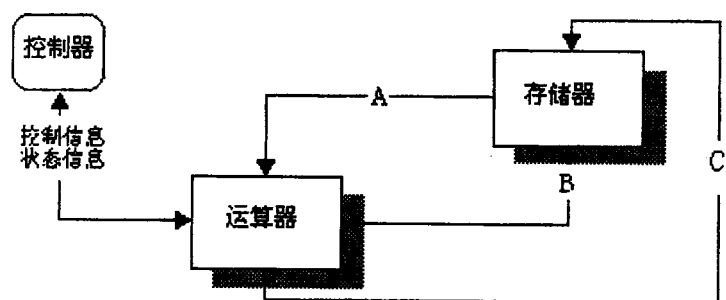
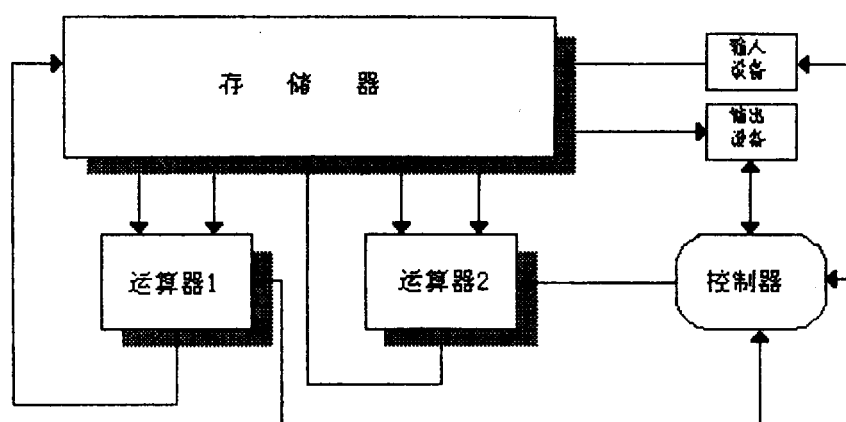


图 一

运算器能完成的运算种类及所需时间如下表所示：

运算种类	运算操作	所需运算时间
1	$C=A+B$	T_{add}
2	$C=A-B$	T_{sub}
3	$C=A*B$	T_{mul}
4	$C=A/B$	T_{div}

在计算复杂的四则混合运算表达式时，运算器就变成了高速计算的“瓶颈”。为了得到更快的运算速度，计算机科学家们设计了一种有两个运算器的并行计算机，它的结构简图如图二所示。



图二

由于两个运算器可以同时进行运算，因此大大提高了整机运算速度。

该并行计算机有以下两种控制指令：

运算指令

OP Time Alu_no Operate_no Address1 Address2 Address3

其中 OP 为运算指令标识符，其后有六个参数：

Time 表示执行该指令的开始时间

Alu_no 表示承担该次运算操作的运算器编号 ($\text{Alu_no} \in \{1, 2\}$)

Operate_no 表示该次运算的运算种类 ($\text{Operate_no} \in \{1, 2, 3, 4\}$)

Address1, Address2 表示待运算的两个源操作数的存储单元地址

Address3 表示该次运算结束后存放运算结果的存储单元地址

结束指令

END Time Address

其中 END 为结束指令标识符，其后有两个参数：

Time 表示整个控制程序的结束时间

Address 表示存放最终计算结果的存储单元地址

每个运算器同一时刻可以执行一条运算指令，结束指令表示控制程序结束。

现在需要你编一个程序，对给定的待计算的表达式，自动生成一段控制程序，使图二所示的并行计算机能够正确计算该表达式的值，并使总运算时间尽量小。

输入

输入文件的第一行为四个不超过 1000 的正整数，依次为 T_{add} 、 T_{sub} 、 T_{mul} 和 T_{div} 。输入文件的第二行为待计算的四则混合运算表达式（长度不超过 255 个字符），表达式中的变量用大写英文字母表示，各变量的初始值依照变量的字母顺序依次存放在地址为 1,2,... 的存储单元中。

输入文件中同一行相邻两项之间用一个或多个空格隔开。

输出

输出文件为完成该表达式计算的最优控制指令段。指令根据其开始执行时间先后依次输出（对于开始执行时间相同的两条指令，输出先后次序不限），每条指令占一行。

输出文件中同一行相邻两项之间用一个空格隔开。

约定

控制程序初始执行时间从 0 时刻开始。

问题中所涉及的各种时间量的单位相同。

存储器的存储单元最多有 1K 个。

由于数据读写时间同运算时间相比较小，可忽略不计。

如果两个运算器同时对同一个存储单元进行操作，则运算器 1 先操作，运算器 2 后操作

评分

程序的得分将取决于其所能找到的最优解与标准最优解相比较的优劣程度。

样例输入

2 2 4 12

$C+(A+B)*C-E/F+F$

样例输出

OP 0 1 1 1 2 6

OP 0 2 4 4 5 8

OP 2 1 1 3 5 7

OP 4 1 3 6 3 10

OP 8 1 1 10 7 11

OP 12 1 2 11 8 12

END 14 12