

# NOI98 第一试试题

## 个人所得税

某国个人所得税法规定，普通公民的主要应纳税收入项目及纳税金额如下：工资、薪金所得。按月计算征税，以每月收入额减除费用 800 元后的余额作为该月应纳税所得额，税率如下表所示：

| 级数 | 月应纳税所得额                | 税率（%） |
|----|------------------------|-------|
| 1  | 不超过 500 元的             | 5     |
| 2  | 超过 500 元~2000 元的部分     | 10    |
| 3  | 超过 2000 元~5000 元的部分    | 15    |
| 4  | 超过 5000 元~20000 元的部分   | 20    |
| 5  | 超过 20000 元~40000 元的部分  | 25    |
| 6  | 超过 40000 元~60000 元的部分  | 30    |
| 7  | 超过 60000 元~80000 元的部分  | 35    |
| 8  | 超过 80000 元~100000 元的部分 | 40    |
| 9  | 超过 100000 元的部分         | 45    |

一次性劳动报酬所得。按次计算征税，每次不超过 4000 元的，减除费用 800 元；4000 元以上的，减除 20% 的费用，余额为应纳税所得额。征税税率如下表所示：

| 级数 | 每次应纳税所得额              | 税率 (%) |
|----|-----------------------|--------|
| 1  | 不超过 20000 元的部分        | 20     |
| 2  | 超过 20000 元~50000 元的部分 | 30     |
| 3  | 超过 50000 元的部分         | 40     |

由上面可以看出，个人工资、薪金及一次性劳动报酬所得都是按照超额累进税率来征税的。超额累进税率将应纳税所得额按数额大小分成若干等级，每一等级规定一个税率，税率依次提高，但每一纳税人的应纳税所得额依照所属等级同时适用几个税率分别计算，将计算结果相加后的总额作为应纳税款。

例如，某人某月工资总额为 3800 元，减去 800 元后，应纳税所得额为 3000 元。其中 1 级 500 元，2 级 1500 元，3 级 1000 元，税率分别为 5%、10%、15%，应纳税总额为  $500 \times 5\% + 1500 \times 10\% + 1000 \times 15\% = 325$ （元）。计算过程如图一所示。

现在需要你编一程序，根据该国某公司的所有职员一年内的各项收入信息（收入项目、收入时间、收入金额）计算该公司所有职员这一年应交纳的个人所得税总额。

| 应纳税所得额 |                | 工资总额 |
|--------|----------------|------|
| 3000   | 第3级<br>(税率15%) | 3800 |
| 2000   | 第2级<br>(税率10%) | 2800 |
| 500    | 第1级<br>(税率5%)  | 1300 |
| 0      | 减除费用           | 800  |
|        |                | 0    |

## 输入

输入文件的第一行为一个正整数  $M$  ( $M \leq 50000$ )，表示该公司的职员总数（职员编号依次为  $1, 2, \dots, M$ ）。接下来的各行每行表示一年内某一个职员的一项收入信息。具体格式如下：

工资、薪金收入信息：PAY 职员编号 收入时间 收入金额

一次性劳务报酬收入信息：INCOME 职员编号 收入时间 收入金额

其中，收入时间格式为：MM/DD，MM 表示月份 ( $1 \leq MM \leq 12$ )，DD 表示日

期 ( $1 \leq DD \leq 31$ ) ; 收入金额是一个正整数 (单位: 元) , 并假设每人每项收入金额小于 100 万元。  
输入文件以字符“#”表示结束。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**

输出文件只有一个正数  $P$  ,  $P$  表示该公司所有职员一年内应交纳的个人所得税总额 (单位: 元) 。

**样例输入**

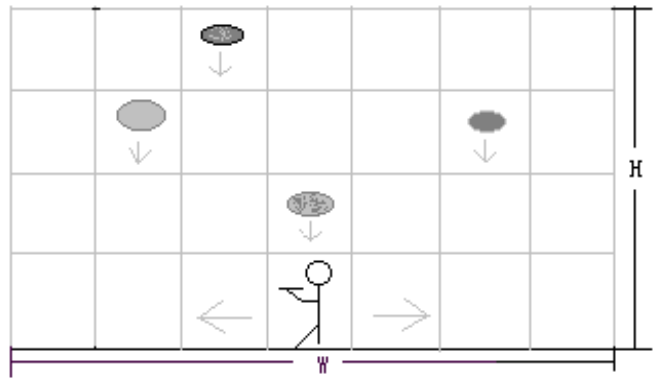
```
2
PAY 1 2/23 3800
INCOME 2 4/8 4010
INCOME 2 4/18 800
PAY 1 8/14 6700
PAY 1 8/10 1200
PAY 2 12/10 20000
#
```

**样例输出**

5476.6

**免费馅饼**

SERKOI 最新推出了一种叫做“免费馅饼”的游戏:游戏在一个舞台上进行。舞台的宽度为  $W$  格, 天幕的高度为  $H$  格, 游戏者占一格。开始时游戏者站在舞台的正中央, 手里拿着一个托盘。下图为天幕的高度为 4 格时某一个时刻游戏者接馅饼的情景。



游戏开始后, 从舞台天幕顶端的格子中不断出现馅饼并垂直下落。游戏者左右移动去接馅饼。游戏者每秒可以向左或向右移动一格或两格, 也可以站在原地不动。

馅饼有很多种，游戏者事先根据自己的口味，对各种馅饼依次打了分。同时，在 8-308 电脑的遥控下，各种馅饼下落的速度也是不一样的，下落速度以格/秒为单位。

当馅饼在某一秒末恰好到达游戏者所在的格子中，游戏者就收集到了这块馅饼。

写一个程序，帮助我们的游戏者收集馅饼，使得所收集馅饼的分数之和最大。

### 输入

输入文件的第一行是用空格隔开的两个正整数，分别给出了舞台的宽度 W（1 到 99 之间的奇数）和高度 H（1 到 100 之间的整数）。

接下来依馅饼的初始下落时间顺序给出了所有馅饼的信息。每一行给出了一块馅饼的信息。由四个正整数组成，分别表示了馅饼的初始下落时刻（0 到 1000 秒），水平位置、下落速度（1 到 100）以及分值。游戏开始时刻为 0。从 1 开始自左向右依次对水平方向的每格编号。

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

输出文件的第一行给出了一个正整数，表示你的程序所收集的最大分数之和。

其后的每一行依时间顺序给出了游戏者每秒的决策。输出 0 表示原地不动、1 或 2 表示向右移动一步或两步、-1 或 -2 表示向左移动一步或两步。输出应持续到游戏者收集完他要收集的最后一块馅饼为止。

### 样例输入

```
3 3
0 1 2 5
0 2 1 3
1 2 1 3
1 3 1 4
```

### 样例输出

```
12
-1
1
1
```

## 软件安装盘

软件安装通常是一件令人头疼的事。软件一般都包括若干个相对独立的部分（称为“组件”），在安装的时候由用户决定安装哪些部分。并且，这些相对独立的组件之间在安装时有一定的先后顺序要求。

由于当代的个人计算机普遍安装了软盘驱动器，所以软件的最流行的载体形式是软盘。然而，由于软盘的容量有限，稍大一些的软件就无法用一张软盘装下。这时，这些软件往往要用很多张软盘来存储。每张磁盘上存储了软件的一

个或多个组件。这些软盘称为软件的安装盘。

由于软件的各个组件分散在不同的软盘上，而在安装时又有一定的先后顺序要求，所以很容易发生要求用户反复换盘的情况。而计算机用户在安装软件的时候，最反感的就是反复在软盘之间切换：找盘、插盘、取盘、找盘、插盘、取盘、…，一切都显得那么琐碎和无序。因此，有必要对软件安装盘的制作提出下述要求：

永远不要让用户将一张磁盘插入两次。更精确地，要求对安装盘从 1 开始顺序编号，使得安装的时候，用户只要按顺序插入磁盘即可。

出于经济的考虑，通常要求安装盘的总数最少。写一个程序，对于给定的软件，制定最优的安装盘方案。

### 输入

输入文件的第一行是一个正整数  $M$  ( $1 \leq M \leq 10^9$ )，给出了每张磁盘的最大容量（字节数）。

输入文件的第二行是一个正整数  $N$  ( $1 \leq N \leq 100$ )，给出了软件的组件数。接下来的  $N$  行每行给出一个组件的详细信息。包括：

① 组件所占的字节数；

② 在安装该组件之前应先安装的组件序号（如有多个组件须先安装，则每个都应列出其序号，若无须先安装其它组件，则该行只含组件所占字节数）。

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

输出文件的第一行给出了最优安装盘方案的软盘数。如果不存在最优安装盘方案，则输出 0。接下来的每一行顺序给出了每张盘上存储的组件的序号。如果一张盘上存储了多个组件，则输出所有这些组件的序号（次序任意），中间用一个空格隔开。

### 样例输入

```
1457664
3
512665
912345 1
832542 1
```

### 样例输出

```
2
1 3
2
```