

## escription

刚刚过去的期中考试中 zyz 同学依稀记得他的准考证号是 37，现在他又将要面临一场比赛，他希望准考证号不出现 37（连续），同时他又十分讨厌 4，所以也不希望 4 出现在准考证号中。。。现在他想知道在 A 和 B 之间有多少合法的准考证号

## Input

包含两个整数，A B

## Output

一个整数。

## Sample Input

【输入样例一】

1 10

【输入样例二】

25 50

## Sample Output

【输出样例一】

9

【输出样例二】

14

【数据规模和约定】

20%的数据，满足  $1 \leq A \leq B \leq 1000000$  。

100%的数据，满足  $1 \leq A \leq B \leq 2000000000$  。

## Description

现在我们的手头有  $N$  个软件，对于一个软件  $i$ ，它要占用  $W_i$  的磁盘空间，它的价值为  $V_i$ 。我们希望从中选择一些软件安装到一台磁盘容量为  $M$  计算机上，使得这些软件的价值尽可能大（即  $V_i$  的和最大）。

但是现在有个问题：软件之间存在依赖关系，即软件  $i$  只有在安装了软件  $j$ （包括软件  $j$  的直接或间接依赖）的情况下才能正确工作（软件  $i$  依赖软件  $j$ ）。幸运的是，一个软件最多依赖另外一个软件。如果一个软件不能正常工作，那么它能够发挥的作用为  $0$ 。

我们现在知道了软件之间的依赖关系：软件  $i$  依赖软件  $D_i$ 。现在请你设计出一种方案，安装价值尽量大的软件。一个软件只能被安装一次，如果一个软件没有依赖则  $D_i=0$ ，这时只要这个软件安装了，它就能正常工作。

## Input

第 1 行：  $N, M$  ( $0 \leq N \leq 100, 0 \leq M \leq 500$ )

第 2 行：  $W_1, W_2, \dots, W_i, \dots, W_n$  ( $0 \leq W_i \leq M$ )

第 3 行：  $V_1, V_2, \dots, V_i, \dots, V_n$  ( $0 \leq V_i \leq 1000$ )

第 4 行：  $D_1, D_2, \dots, D_i, \dots, D_n$  ( $0 \leq D_i \leq N, D_i \neq i$ )

## Output

一个整数，代表最大价值。

## Sample Input

```
3 10
5 5 6
2 3 4
0 1 1
```

## Sample Output

食堂 128M 1s dining.cpp

## Description

小 F 的学校在城市的一个偏僻角落，所有学生都只好在学校吃饭。学校有一个食堂，虽然简陋，但食堂大厨总能做出让同学们满意的菜肴。当然，不同的人口味也不一定相同，但每个人的口味都可以用一个非负整数表示。由于人手不够，食堂每次只能为一个人做菜。做每道菜所需的时间和前一道菜有关的，若前一道菜的对应的口味是  $a$ ，这一道为  $b$ ，则做这道菜所需的时间为  $(a \text{ or } b) - (a \text{ and } b)$ ，而做第一道菜是不需要计算时间的。其中， $\text{or}$  和  $\text{and}$  表示整数逐位或运算及逐位与运算，C 语言中对应的运算符为 “|” 和 “&”。学生数目相对于这个学校还是比较多的，吃饭做菜往往就会花去不少时间。因此，学校食堂偶尔会不按照大家的排队顺序做菜，以缩短总的进餐时间。虽然同学们能够理解学校食堂的这种做法，不过每个同学还是有一定容忍度的。也就是说，队伍中的第  $i$  个同学，最多允许紧跟他身后的  $B_i$  个人先拿到饭菜。一旦在此之后的任意同学比当前同学先拿到饭，当前同学将会十分愤怒。因此，食堂做菜还得照顾到同学们的情绪。现在，小 F 想知道在满足所有人的容忍度这一前提下，自己的学校食堂做完这些菜最少需要多少时间。

## Input

第一行包含一个正整数  $C$ ，表示测试点的数据组数。每组数据的第一行包含一个正整数  $N$ ，表示同学数。每组数据的第二行起共  $N$  行，每行包含两个用空格分隔的非负整数  $T_i$  和  $B_i$ ，表示按队伍顺序从前往后的每个同学所需的菜的口味和这个同学的忍受度。每组数据之间没有多余空行。

## Output

包含  $C$  行，每行一个整数，表示对应数据中食堂完成所有菜所需的最少时间。

## Sample Input

```
2
5
5 2
4 1
12 0
3 3
2 2
2
5 0
4 0
```

## Sample Output

```
16
1
```

## HINT

对于第一组数据：同学 1 允许同学 2 或同学 3 在他之前拿到菜；同学 2 允许同学 3 在他之前拿到菜；同学 3 比较小气，他必须比他后面的同学先拿菜…… 一种最优的方案是按同学 3、同学 2、同学 1、同学 4、同学 5 做菜，每道菜所需的时间分别是 0、8、1、6 及 1。

### 【数据规模和约定】

对于 30% 的数据，满足  $1 \leq N \leq 20$ 。

对于 100% 的数据，满足  $1 \leq N \leq 1,000$ ,  $0 \leq T_i \leq 1,000$ ,  $0 \leq B_i \leq 7$ ,  $1 \leq C \leq 5$ 。

存在 30% 的数据，满足  $0 \leq B_i \leq 1$ 。

存在 65% 的数据，满足  $0 \leq B_i \leq 5$ 。

存在 45% 的数据，满足  $0 \leq T_i \leq 130$ 。