

event
sponsor

The 1st ACM-ICPC Thailand National Programming Contest 2009

11 August 2009

Hosted by Prince of Songkla University, Phuket Campus

- There are **8 problems** (A-H) to solve within 300 minutes (5 hours).
- Solve as many problems as you can in an order of your choice.
- Use **C** or **C++** or **Java** to program at your convenience for any problems.
- Input and output of each program are **standard input** and **output**.



Problem A. String Transformation

กำหนดกฎในการแปลงข้อความหนึ่งไปเป็นอีกข้อความหนึ่งดังนี้ :

- 't' \Rightarrow 'i' (เปลี่ยน t เป็น i)
- {'a' - 'z'} \Rightarrow 'm' (เปลี่ยน 'a' หรือ 'b' หรือ 'c', ..., หรือ 'z' เป็น 'm')
- 's' \Rightarrow 'i' (เปลี่ยน 's' เป็น 'i')
- {'a' - 'z'} \Rightarrow " (สามารถลบอักษร 'a' หรือ 'b' หรือ 'c', ..., หรือ 'z' ได้)
- 't' \Rightarrow 'r' (เปลี่ยน 't' เป็น 'r')
- {'a' - 'g'} \Rightarrow 'h'
- 'o' \Rightarrow 'a'
- 't' \Rightarrow 's'
- {'t' - 'z'} \Rightarrow 'u'

เซตของตัวอักษรเขียนเป็นช่วง {'a' - 'd'} เป็นเซตของตัวอักษร {'a', 'b', 'c', 'd'} ให้เขียนโปรแกรมในการอ่านข้อความ 2 ชุดและตรวจสอบว่าข้อความแรกสามารถแปลงไปเป็นข้อความที่ 2 ได้หรือไม่ โดยใช้กฎการแปลงข้อความที่กำหนดให้ก่อนหน้านี้ ถ้าสามารถแปลงได้ให้แสดงข้อความ "YES" ถ้าไม่ได้ ให้แสดงข้อความ "NO"

Input

อินพุตเป็นข้อความที่ต้องการเปรียบเทียบทีละสองบรรทัด นั่นคือทำการเปรียบเทียบ บรรทัดที่ i และ บรรทัดที่ i+1 โดยแต่ละข้อความยาวสูงสุดไม่เกิน 256 ตัวอักษร อินพุตถือว่าสิ้นสุดเมื่อข้อความนั้นเป็น "000" (ศูนย์ตามตัวติดกัน)

Output

เป็นผลลัพธ์จากการทดสอบการแปลงข้อความที่ได้โดยที่แสดงข้อความ "YES" ถ้าข้อความแรกสามารถแปลงเป็นข้อความที่สองได้ และถ้าแปลงไม่ได้ ให้แสดงข้อความ "NO"

Sample Input	Sample Output
ttsze {	YES
isium {	NO
a {	NO
x }	YES
xyza	YES
aa	
bbbbbbtt :	
uu	
f	
f	
000	



Problem B. Minimum Swap

การจัดเรียงข้อมูลเป็นขั้นตอนวิธีที่ถูกเรียกใช้บ่อยครั้งในการพัฒนาซอฟต์แวร์ซึ่งมีทั้งแบบการจัดเรียงจากน้อยไปมากและทั้งจากมากไปน้อยซึ่งในการจัดเรียงข้อมูลจะต้องมีการเปรียบเทียบและการสลับที่ (swap) ของข้อมูลเกิดขึ้น ให้ผู้เข้าแข่งขันหาจำนวนครั้งในการสลับที่ที่น้อยที่สุดของการจัดเรียงข้อมูลจากน้อยไปมาก

Input

แต่ละบรรทัดประกอบด้วยชุดของตัวอักษรตัวเล็ก (lowercase) ที่ติดกันและจะไม่มีตัวอักษรที่ซ้ำกันในแต่ละชุดข้อความ อินพุตสิ้นสุดด้วยเมื่อข้อความบรรทัดนั้นเป็น "000"

Output

ผลลัพธ์แต่ละบรรทัดให้ค่าจำนวนครั้งต่ำที่สุดที่ใช้ในการสลับที่ข้อมูลในการเรียงลำดับข้อมูลของตัวอักษรจากน้อยไปมาก

Sample Input	Sample Output
abc	0
xyz	0
cba	1
acb	1
bdca	2
fedcba	3
000	

**Problem C. Tourist Bus Organizing****การจัดรถทัวร์**

บริษัท Sawadee Phuket Tour รับจัดรถทัวร์ให้กลุ่มนักท่องเที่ยว เป็นทัวร์วันเดียว (one-day trip) โดยมีรถเช่าให้บริการ ซึ่งมีรายละเอียดและค่าใช้จ่ายดังตาราง

ประเภทรถ	จำนวนรถ (คัน)	จำนวนผู้โดยสาร (ไม่รวมคนขับและไกด์)	เบี่ยงเลี้ยงคนขับรถ และไกด์ (บาท/คัน)	ราคาค่าเช่า (บาท/วัน)	ค่าเชื้อเพลิง (บาท/km)
1.รถตู้ (van)	15	12	1200	800	11
2.รถมินิบัส (minibus)	10	20	1500	1000	14
3.รถโค้ช (coach)	8	30	1800	2000	20
4.รถบัส (autobus)	8	50	2000	2500	25

การจัดทัวร์ครั้งหนึ่งๆ กลุ่มรถทัวร์จะเดินทางไปในเส้นทางเดียวกัน(ระยะทางคิดเป็นกิโลเมตร) บริษัทจะต้องจัดจำนวนรถแต่ละประเภทให้เหมาะสมกับจำนวนรถทัวร์และระยะทาง โดยคิดให้มีค่าใช้จ่ายน้อยที่สุด และรถหนึ่งคันจะใช้น้ำมันในหนึ่งวัน

Input

โปรแกรมรับจำนวนทัวร์ (n) ที่ต้องจัดในหนึ่งวัน จากนั้นรับจำนวนรถทัวร์ (np), $5 \leq np \leq 1000$ และ ระยะทางเป็นกิโลเมตร (nkm), $10 \leq nkm \leq 2000$ ของแต่ละทัวร์ จำนวนรถทัวร์รวมต่อวันมีไม่เกิน 1000 คัน

Input: บรรทัดที่ i มีค่า n ($1 \leq n \leq 30$) คือ จำนวนทัวร์ที่ต้องจัดในหนึ่งวัน

บรรทัดที่ i ($2 \leq i \leq n+1$) มีจำนวน np และ nkm ของทัวร์ ที่ i-1

Output

โปรแกรมต้องคำนวณหาจำนวนรถแต่ละประเภทที่ต้องใช้สำหรับแต่ละทัวร์โดยให้มีค่าใช้จ่ายที่น้อยที่สุด แล้วแสดงจำนวนรถแต่ละประเภทที่ต้องใช้ตามลำดับ และค่าใช้จ่ายของทัวร์นั้น

Output: บรรทัดที่ j ($1 \leq j \leq n$) แสดงจำนวนรถแต่ละประเภทตามลำดับและค่าใช้จ่ายของทัวร์ที่ j

หมายเหตุ ให้จัดรถให้กับทัวร์ตามลำดับ ทัวร์ลำดับหลังจะใช้รถที่มีเหลืออยู่เท่านั้น หากทัวร์ใดมีรถไม่เพียงพอกับรถทัวร์ ไม่สามารถจัดได้ จะได้จำนวนรถและค่าใช้จ่ายเป็นศูนย์ทั้งหมด

Sample Input	Sample Output
2	0 1 0 0 6700
16 300	0 2 2 8 115600
500 250	

**Problem D. Happy Prime Number**

A *Happy Number* ☺ can be defined as follows. From a positive integer n , calculate the sum of square of each digit of n . Then from that sum, repeat the same process over and over again. This cycle terminates if and only if there is 1 in the sequence. Hence, we call the number n a *happy number* if it generates a finite sequence. Otherwise, the endless cycle occurs (1 never appears in the sequence). We may call the number generating an endless cycle an *unhappy number* ☹. Observe the following examples:

700 is a happy number

$$7^2 + 0^2 + 0^2 = 49$$

$$1^2 + 3^2 + 0^2 = 10$$

$$4^2 + 9^2 = 97$$

$$1^2 + 0^2 = 1$$

$$9^2 + 7^2 = 130$$

2 is not a happy number

$$2^2 = 4$$

$$3^2 + 7^2 = 58$$

$$1^2 + 4^2 + 5^2 = 42$$

$$4^2 = 16$$

$$4^2 = 16$$

$$5^2 + 8^2 = 89$$

$$4^2 + 2^2 = 20$$

$$1^2 + 6^2 = 37$$

$$8^2 + 9^2 = 14$$

$$2^2 + 0^2 = 4$$

... and never terminates

A *Prime Number* is an integer greater than 1 that can be divided only by 1 and itself. Here are some prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, ...

A *Happy Prime Number* is a prime number which also satisfies the happy number condition such as 7, 13, 19, ...

Your task is to write a program to show all happy prime numbers less than or equal to a given number n . ($10 \leq n \leq 1000000$)

Input and Output

A positive number n is the only input of the program and the program prints all happy number in ascending order, one number in a line.

Sample Input	Sample Output
20	7 13 19

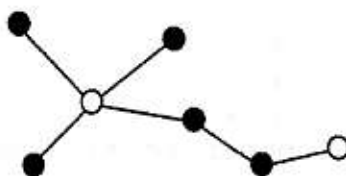


Problem E. Internet Café

The city of IC is very well planned. There are N intersections and $N - 1$ roads. Each road directly joins two intersections at its ends. Also, it is possible to travel from any intersection to another intersection using these roads.

Intersections are good locations for Internet cafe. The city plans to build as many Internet Cafe's at these intersections, however, too many of them located very closely will make the competition too fierce. Therefore, there is a regulation that each Internet cafe can not *directly* connected to more than one other Internet cafe through any road.

Below is a map of an example city. There are 7 intersections, all shown as circles. Roads are shown as lines connecting these intersections. In this case the maximum number of Internet cafe's that can be opened is 5 (shown as black circles).



Your task is to write a program that given the map of the city and determines the maximum number of Internet cafe's that can be opened in this city.

Input

The first line of the input contains an integer C ($1 \leq C \leq 7$) denoting the number of data sets. The first line for each data set contains an integer N ($1 \leq N \leq 1,000$) the number of intersections. The intersections are numbered from 1 to N . The next $N-1$ lines specify how roads connect these intersections. Each of these lines contains two integer A and B that says that there is a road directly joining intersections A and B .

Output

Each of C lines contains an integer which is the maximum number of Internet cafe that can be opened for each data set.

Sample input	Sample output
2	5
7	3
1 2	
2 3	
4 2	
2 5	
5 6	
6 7	
4	
1 2	
2 4	
4 3	



Problem F. Wrestling Teams Selection

The *Thai Wrestling Association* must divide its members into two teams.

Each person must be on one team or the other; the number of people on the two teams must not differ by more than 1; the total weight of the people on each team should be as nearly equal as possible.



Input

The first line of input contains n the number of people in the association, and n lines follow. The first line gives the weight of person 1; the second the weight of person 2; and so on. Each weight is an integer between 1 and 450. There are at most 100 people in the association.

Output

Your output will be a single line containing 2 numbers: the total weight of the people on one team, and the total weight of the people on the other team. If these numbers differ, give the smaller first.

Sample Input

```
3
100
90
200
```

Sample Output

```
190 200
```




Problem G. My Ancestor

Your great great ... great grandfather 'Tiger' was a famous *Muay Thai (Thai Boxing)* master. He was renowned for his trademark '*Teh Chiang*' (*Diagonal Kick*). This skill has surely well preserved within your family as you have inherited his *Teh Chiang* **plus** all other skills from your ancestors and many more, e.g. *Chok* (*Punch*), *Tee Sok* (*Elbow*), *Teh* (*Kick*), *Tee Kao* (*Knee*), *Teep* (*Foot-thrust*) techniques.

Your family has a very good Muay Thai training programme so that all children in your family always have **strictly greater level** of skill than their parents. Given a family structure (only males are listed) and the skill level of each family member, you want to know which of your **greatest ancestor** (the man closest to 'Tiger' in family structure and may include yourself) is the first man to reach some level of skill.



Input and Output

The input contains multiple test cases; please process each test case until end of file. There is **NO** blank line between test cases.

The first line of each test case contains two integers: N, Q .

N denotes the number of person in your family. $1 \leq N \leq 200000$.

Q denotes the number of queries. $1 \leq Q \leq 100000$.

Then, $N-1$ lines appear, line 1 until line $N-1$.

The i -th line contains two integers: pr_i and lv_i . $0 \leq pr_i \leq N-2$; $1 \leq lv_i \leq 2^{31}-1$.

This means person i has skill level lv_i and has pr_i as his parent.

Finally, Q lines appear, line 1 until line Q .

The i -th line contains two integers: $person$ and P . $0 < person \leq N-1$; $1 \leq P \leq lv_{person}$.

For each query, please print the index of greatest ancestor of $person$ with skill level **greater than or equal to P** .

Tiger has index = 0, $pr_0 = -1$, and $lv_0 = 1$, but there will be no query with $person = 0$, so this boundary case can be ignored.



Sample Input and Sample Output

Sample input	Sample output
7 4	0
0 3	2
0 4	2
0 2	5
1 4	0
2 7	2
2 10	2
5 1	5
5 3	
5 4	
5 7	
7 4	
0 3	
0 4	
0 2	
1 4	
2 7	
2 10	
5 1	
5 3	
5 4	
5 7	

Explanation

Observe that there is no blank line between the outputs of these two identical test cases!



Problem H. Biggest Space

An exhibition hall has an empty rectangular space of length L and width W for rent. One can view the space as an rectangle on an x-y co-ordinate plane such that $(0,0)$, $(L,0)$, $(0,W)$, and (L,W) are four corners of the rectangle.

There is a simple rule for renting. Each company must rent a single contiguous area which can be either circular or rectangular. No two areas can overlap, i.e., they cannot have *non-empty* intersection. Also, to simplify the cost calculation, each vertex of each area must have all integer co-ordinates.

This space has already be rented out to N companies. Your company also wants to rent a rectangular area in this hall and asks you to write a program to find the biggest empty rectangular area.

Input

The first line contains an integer C ($1 \leq C \leq 11$) denoting the number of data sets. Each data set starts with a line containing 3 integer L W and N ($1 \leq L \leq 30$; $1 \leq W \leq 30$; $1 \leq N \leq 20$). After that, there are N lines describing each rented area. Each line starts with an integer S which can be 0 or 1.

When S is 1, the area is a rectangle and 4 integers X_1 Y_1 X_2 Y_2 ($0 \leq X_1 < X_2 \leq L$; $0 \leq Y_1 < Y_2 \leq W$) that follow S specify two corners of the area, which are (X_1, Y_1) and (X_2, Y_2) .

When S is 0, the area is a circle and 3 integers X Y R that follow S specify that the circle is centered at (X, Y) ($0 < X < L$; $0 < Y < W$) and has radius R ($R > 0$).

Output

There are C lines, one line for each data set. For each data set, your program should output the maximum area of the largest empty rectangle.

Sample input	Sample output
3	36
10 10 2	9
0 3 3 3	2
1 9 9 10 10	
10 10 3	
0 4 4 4	
1 0 8 6 10	
1 8 0 10 6	
10 10 1	
0 5 5 5	