

# Caesar Cypher (caesar)

Julius Caesar has just written the new war declaration against Sparta. To prevent the enemies from reading this precious document the Emperor has ordered to encrypt it using his cypher.

The Caesar Cypher consists of changing each letter of the message with next k-th letter of the alphabet (wrapping to a after z). k is called the key of the cypher and it can be used to decrypt the message simply by re-encrypting it with -kas the key.

For example if the message was pizza and the key k = 3 the encrypted message would be slccd.



Unfortunately the official who was supposed to encrypt the messages messed it up. He put the war declaration messages with other unrelated messages. Luckily for him he noticed that if he splits the messages into piles all with the same message (maybe encrypted with different keys), the taller is the one!

Help him to find the size of the largest pile all with the same message.

Among the attachments of this task you may find a template file caesar.\* with a sample incomplete implementation.

#### Input

The first line contains two integers N and D, where N is the number of messages, D is the length of each message. The next N lines contain a single word each.

### Output

You need to write a single line with an integer: the size of the largest pile.

### Constraints

- $1 \le N \le 10\,000.$
- $1 \le D \le 1000$ .
- Each message is a single word with only lower case ASCII letters.



caesar • EN

### Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases. These test cases are such that:

• in 50% of them,  $N \le 1000$ ,

#### **E**xamples

stdin/input.txt	stdout/output.txt
5 4 aaaa bbbb abab fgfg	3
hhhh	
5 5 pizza mafia cacca puzza pizzo	1

### Explanation

In the **first example** there are 2 piles:

- $\bullet\,$ aaaa, bbbb and hhhh
- abab, fgfg

The largest is the first so the answer is 3. Note that, assuming **aaaa** is the unencrypted message, **bbbb** can be decrypted with k = 1 and **hhhh** with k = 7.

In the **second example** each message is unique and cannot be transformed into any others. So there are 5 piles each with one message.



copac • EN

# Wonderful Grove (copac)

Silviu the lumberjack, who lives in "The Wonderful Grove", is well known for his ability and is ready to tackle another challenging task. Winter is coming and a big tree needs to be cut.

Silviu is carefully studying the structure of the tree, taking precise notes about each branch and its connections. He has labelled all the points of connection between two or more branches with numbers from 1 to N.

He has opted for making a single cut (after all, every cut requires physical effort...), getting rid of the branches con-



juncting there and transporting the remaining smaller pieces. The company selected for transport is going to accept a part of the tree only if it is smaller (or equal) than half of the size of the original tree. The size of a (part of a) tree is merely the number of the conjunction points in it.

At which points can Silviu make his cut to satisfy the requirement from the transport company?

 $\square$  Among the attachments of this task you may find a template file copac.\* with a sample incomplete implementation.

#### Input

The first line contains the only integer N. The following N-1 lines contain two integers  $X_i$  and  $Y_i$  each, meaning that there is a branch between those two points.

### Output

You need to write a line for every conjuction point that can be chosen for the cut. If multiple points respect the constraint, they must be sorted in ascending order.

### Constraints

- $1 \le N \le 100\,000.$
- $1 \leq X_i, Y_i \leq N$  for each  $i = 0 \dots N 2$ .
- It is guaranteed that at least one answer exists.

### Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases.



copac • EN

### Examples

stdin/input.txt	stdout/output.txt				
7	2				
	3				
1 2					
1 3					
2 4					
3 5					
3 6					
3 7					

## Explanation

In the **sample case** the only feasible solution is to make a cut at point 3. In that way the original tree is divided in four pieces (one of size 3 and three of size 1).



# **Decimal Fractions (fraction)**

Archimedes has just discovered a new concept: the *fraction*, where an integer number (numerator) is divided by another integer number (denominator). In particular, he is now focusing on the simplest kind of fractions, called *unit fraction*, where the numerator is exactly 1.

1/99800	L = 0.0	(000-	001 0	002 0	03	004	005	006	007	008	009	010	011	012	013	014	015	016	017	018	019
020 021 022	2 023 02	24 025	026	027 0	28 02	29 03	0 031	032	033	034 (	35 0	36 03	7 032	8 039	040	041	042	043	044	045	046
047 048 049	050 051	052 05	3 054	055 0	56 057	7 058	059 06	0 061	062 C	63 06	4 065	066 0	67 068	069 0	070 07	1 072	073 0	74 075	076 0	77 07	8 079
080 081 082 0	83 084 0	85 086	०४७ ०४१	४ ०४७ ०	90 091	1 092 0	93 094	095 0	96 097	098 09	9 100 1	101 102	103 10	4 105	106 10	07 108	109 11	o 111 1	12 113	114 1	15 116
117 118 119 120	121 122 123 3 164 165 1	124 125	126 127 : 8 169 170	128 129 0 171 172	130 131	1 1 <i>3</i> 2 13 4 175 13	33 134 1 76 177 17	.35 136 8 179 15	137 138	r 139 14 82 183 1	0 141 1 84 185	186 185	144 14	-5146 919019	147 14	8 149 1	50 151 196 19	152 153 97 198 19	154 15 9 200 2	5 156 1; 01 202 2	57 158 03 204
205 206 207 208 20	09 210 211 2	12 213 214	4 215 216	217 218	219 220	221 222	223 224	225 226	227 228	229 230	231 232	233 234	235 236	237 23	8 239 24	+0 241 2	42 243	244 24:	5 246 24	7 248 2	49 250
251 252 253 254 255	256 257 258	259 260 20	1 262 263	8 264 265	266 267	268 269	270 271	272 273 2	74 275 2	76 277 27	8 279 28	0 281 28	283284	285 28	6 287 28	18 289 29	0 291 29	2 293 294	295 296	297 298	299 300
348 349 350 351 3	52 353 354 3	366 366 36	7 358 359	9 360 361	362 363	3 364 36	5 366 36	7 368 30	59 370 37	1 372 37	3 374 37	5 376 377	378 379	380 38	1 382 38	3 384 3	85 386 3	387 388	389 390	391 392	393 394
395 396 397 398 399 4 443 444 445 446	00 401 402 4 447 448 44	403 404 4 49 450 451	05 406 40 452 453	07 408 40 454 455 ·	09 410 4 456 457	458 459	-13 414 4 460 461	462 463	464 46	+19 420 4 5 466 467	21 422 4	23 424 4 3 470 471	25 426 41 . 472 473	7 428 4 474 475	29 430 4 476 477	-31 432 4 478 479	33 434 4 480 48	435 436 1 482 48	437 438 3 484 4	439 440 85 486 4	) 441 442 487 488
489 490 491 492 4	93 494 495 4	+96 497 49	8 499 500	501 502 6	503 504	505 506	507 508 5	09 510 51	1 512 512	514 515	516 517 5	18 519 52	0 521 522	523 524	525 526 :	527 528 52	9 530 53	1 532 533	534 535	536 537	538 539
591 592 593 594 595 596	+ 545 546 5 597 598 599 (	47 948 94 500 601 602	603 604 6	992 993 99 05 606 607	608 609	610 611 6	12 613 614	615 616	563 564 617 618 6:	19 620 621	622 623 61	9 570 571 14 625 626	627 628 6	14 979 9 29 630 63	1 632 633	634 635 6	536 637 6	: 783 784 38 639 64	• 585 589 • 641 64	643 644	645 646
647 648 649 650 651 6 703 704 705 706 707 7	52 653 654 65 08 709 710 711	5 656 657 6 712 713 714	58 659 660 - 715 716 71	661 662 66 17 718 719	53 664 66 720 721 7	65 666 661 22 723 724	7 668 669 ( + 725 726 7	570 671 67 27 728 729	2 673 674 9 730 731	675 676 67 732 733 73	7 678 679 4 735 736	680 681 6 737 738 7	82 683 68 39 740 74	4 685 68 1 742 743	6 687 688 744 745	8 689 690 746 747 7	691 692 6° 48 749 7:	93 694 69: 50 751 752	: 696 697 753 754 :	698 699 70 155 756 75	00 701 702 7 758 759
760 761 762 763 764 76	5 766 767 768 22 823 824 82	769 770 771	772 773 774	4 775 776 7 31 832 833	77 778 77	9 780 781 836 837 8	782 783 °	784 785 7 841 842	86 787 78 843 844 :	8 789 790 845 846 81	791 792 79 47 848 84	3 794 795 9 860 861 1	796 797 79	8 799 800 + 855 856	801 802 857 858 8	803 804 59 860 861	805 806 1	807 808 8 864 865 86	09 810 81 6 867 868	1 812 813	814 815
874 875 876 877 878 879 935 936 937 938 939 940 94	880 881 882 8 1 942 943 944	945 946 947	886 887 81 948 949 95	88 889 890 0 951 952 95	891 892 8 3 954 955	893 894 89 956 957 95	8 896 897 8 959 960 9	898 899 90 61 962 963 9	0 901 902 9 964 965 960	903 904 901 967 968 96	906 907 9 9 970 971 9	08 909 910 72 973 974 9	911 912 913 975 976 977 9	914 915 9 78 979 980	16 917 918 981 982 9	919 920 921 83 984 985	922 923 92 986 987 9	24 925 926 88 989 990	927 928 929 991 992 99	930 931 9. 994 995 9	32 933 934 96 997 999)

Figure 1: A remarkable unit fraction: every three-digit number is in the cycle... except for 998, of course.

Until now, Archimedes managed to compute the decimal representation of all the unit fractions with denominator up to ten, and noticed that some of them contain a *recurring cycle* of digits.

1/2 = 0.5 1/3 = 0.(3) 1/4 = 0.25 1/5 = 0.2 1/6 = 0.1(6) 1/7 = 0.(142857) 1/8 = 0.125 1/9 = 0.(1) 1/10 = 0.1

Where 0.1(6) means 0.166666..., and has a 1-digit recurring cycle. Archimedes is particularly excited by long recurring cycles, like the one of 1/7 which is 6 digits long, and would like to discover an even longer one with denominator up to N; but is getting tired of computing all those decimals. Help Archimedes in his quest, by finding the largest number d < N such that 1/d contains the longest possible recurring cycle in its decimal fraction part!

Among the attachments of this task you may find a template file fraction.\* with a sample incomplete implementation.

#### Input

The first and only line contains a single integer N.

### Output

You need to write a single line with an integer: the largest d < N with a longest recurring cycle.



•  $2 \le N \le 1\,000\,000.$ 

### Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases. These test cases are such that:

- in 50% of them,  $N \le 1000$ ,
- in 80% of them,  $N \leq 30\,000$ .

stdin/input.txt	stdout/output.txt
10	7
20	19



# Word Game (jumps)

James Joyce is composing a new mind-blowing masterpiece. In his next work, two words  $w_1$  and  $w_2$  will be allowed to be adjacent only if they are *close enough*. We define  $w_2$  to be close enough to  $w_1$  (or that  $w_1$  jumps to  $w_2$ ) if one of the following conditions is verified:



- $w_2$  is obtained from  $w_1$  by deleting one letter.
- $w_2$  is obtained from  $w_1$  by replacing one of the letters in  $w_1$  by some letter that appears to its right in  $w_1$  and which is also to its right in alphabetical order.

For example, we can jump from abbca to abca by deleting the second (or the third) letter. We can jump from aabca to abbca by replacing the a in the second position by the letter b that appears to the right of the a in aabca and which is also to its right in alphabetical order. On the other hand, we cannot jump from abbca to aabca since we would need to replace the b in the second position by a, but a is to the left of b in alphabetical order.

You will be given a collection of words W, which James want to put in his next masterpiece. Your task is to find the length of the longest sequence  $w_1, w_2, \ldots$  of distinct words from W such that we may jump from  $w_1$  to  $w_2$ ,  $w_2$  to  $w_3$  and so on. We call this the *jumping number* for this set.

For example, if

 $W = \{abacd, bcdada, dd, abcd, bcdd, adcd, addd, aa, ccd, add, ad\}$ 

then, the jumping number is 7 corresponding to the sequence

abacd, abcd, adcd, addd, add, ad, dd.

Among the attachments of this task you may find a template file jumps.\* with a sample incomplete implementation.

#### Input

The first line contains the only integer N indicating the number of words in the input. The following N lines contain one word  $W_i$ .

### Output

The output should be a single integer, indicating the *jumping number* of the given set of words.



- $1 \le N \le 100$ .
- The length of each word is at most 10.
- Each word is formed by lowercase alphabetic letters (a-z).

#### Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases.

stdin/input.txt	stdout/output.txt
11	7
abacd	
bcdada	
dd	
abcd	
bcdd	
adcd	
addd	
aa	
ccd	
add	
ad	



# Fractal Painting (painting)

Ever since Marcus (a famous Dutch abstractionist) heard of fractals, he made them the main topic of his canvases. Every morning, the artist takes a piece of graph paper and starts making a model of his future canvas, by selecting a rectangular area of  $N \times N$  squares and painting some of these squares black. Then, he takes a clean square piece of paper and paints the fractal using the following algorithm:

- 1. The paper is divided into  $N^2$  identical squares and some of them are painted black according to the model.
- 2. Every square that remains white is divided into  $N^2$  smaller squares and some of them are painted black according to the model.
- ÷
- K. Repeat step 2.

At the end of the process, a canvas consisting of  $N^K \times N^K$  black and white squares is produced.



Unfortunately, this tiresome work demands too much time from his painting genius. Marcus has been dreaming of making the process automatic to move to making 3D or even 4D fractals: help him!

Among the attachments of this task you may find a template file **painting**.\* with a sample incomplete implementation.

#### Input

The first line contains integers N, K. Each of the following N lines contains N symbols that determine the model, where:

- '.' represents a white square;
- '\*' represents a black square.

#### Output

You need to produce an  $N^K \times N^K$  matrix representing what a picture should look like after K steps of the algorithm.



- $2 \le N \le 10$ .
- $\bullet \ N^K \leq 17\,000$

## Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases. These test cases are such that:

- in 40% of them, N = 2,
- in 70% of them,  $N \leq 4$ .

stdin/input.txt	stdout/output.txt
2 3	.*****
.*	*****
	.*.****
	****
	.***.**
	****
	.*.*.*
3 2	.*.***.*.
.*.	****
***	.*.**.*.
.*.	****
	****
	****
	.*.***.*.
	****
	.*.***.*.



subset • EN

# Number Game (subset)

Rich Uncle Pennybags is getting tired of playing Monopoly, hence developed a new exciting *number game*. In this game, V tokens are randomly extracted, each depicting a distinct positive integer. Each player then has to select a subset S such that every pair  $(S_i, S_j)$  of elements in this subset satisfies:  $S_i \% S_j = 0$ or  $S_j \% S_i = 0$ . The player who picks up the bigger subset wins!



Rich Uncle Pennybags is not used to lose a game: help him find the largest allowed subset!

Among the attachments of this task you may find a template file subset.\* with a sample incomplete implementation.

#### Input

The first line contains the only integer N indicating the cardinality (number of elements) of the set V. The following line contains N integers, the elements  $V_i$  of the set.

#### Output

The output should be a single integer, indicating the *largest cardinality* of a subset S of V such that the condition is satisfied. If there are multiple solution, output any of them.

#### Constraints

- $1 \le N \le 1000$ .
- $V_i < 2^{32}$  for any i.

#### Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases.

stdin/input.txt	stdout/output.txt					
5 1 2 4 8 5	4					



# UEFA Coefficient (uefa)

Everyone in the world knows Andrew Wiles' addiction to football. Recently, a strange number has been calculated for each country: the "UEFA coeffecient". This coefficient is so important because it determins how many teams for each country are allowed to compete in the world's most prestigious tournament, the UEFA Champions League.

Andrew loves playing with numbers and computers, but he's a bit skeptical about *how* that coefficient is computed by UEFA and would prefer to calculate it himself. He gathered a list of N teams, each with its country, and a list of Mresults of played matches.

The rules to calculate a team's coefficient are quite straightforward:

- Every team gets *one* point for every match it has played;
- Additional *three* points are awarded to the winner (i.e., the team with the highest number of goals scored). If the match ends with a draw, *one* point is awarded to both teams.



The coefficient associated with a country is the sum of all coefficients of teams belonging to it. Help Andrew writing a program that computes the final ranking, ordered by decreasing coefficient, with countries and their global coefficients!

Among the attachments of this task you may find a template file uefa.\* with a sample incomplete implementation.

#### Input

The first line contains integers N and M. The following N lines contain two strings each:  $T_i$  and  $C_i$ , respectively the name of the *i*-th team and its country. The subsequent M lines contain two strings  $(T_{home}$  and  $T_{away})$  and two integers representing the score of that match (i.e. the number of goals, respectively  $G_{home}$  and  $G_{away}$ ).

### Output

You need to write K lines, one for every country. Each of them should contain a string and an integer: respectively, the name of the country and its UEFA coefficient.



- $2 \le N \le 1\,000.$
- $1 \le M \le 10\,000.$
- Each team name  $T_i$  and each country  $C_i$  is a string of at most 10 lowercase letters of the English alphabet.
- A team never plays a match against itself.
- Integers  $G_{home}$  and  $G_{away}$ , the number of goals, are both non negative.
- Whenever two countries share exactly the same coefficient, they should be ordered lexicograpically.

#### Scoring

Your program will be tested against several test cases, and your score will proportional to the number of correctly solved test cases. These test cases are such that:

• in 50% of them, there is only one country (i.e., K = 1).

#### Examples

stdin/input.txt	stdout/output.txt
4 4 inter italy milan italy bayern germany psg france inter milan 1 0 bayern inter 4 1 milan inter 2 2 milan psg 1 0	italy 14 germany 4 france 1
3 3 inter italy milan italy juventus italy inter milan 3 3 juventus inter 0 1 inter juventus 0 3	italy 14

#### Explanation

In the **first sample case** the leading country is Italy with 14 as coefficient, given from 6 points for playing matches, 3 points for Inter winning the first match, 2 points for the draw in the third one, 3 points for Milan winning the fourth match.

In the **second sample case** every team is Italian, thus Italy is the only country appearing in the ranking.