ACM International Collegiate Programming Contest 2002-2003
Northeastern European Regional Programming Contest
Eastern subregion
Second Round

In all tasks read input data from the file INPUT.TXT and write results to the file OUTPUT.TXT.
The format of the input file corresponds to the specification and additional check is not needed.
The new-line character ends each line, including the last one.
Time limit is given for the processor Celeron 500 MHz.

1. Dividing into Groups	5 sec
2. The Travel of the Queen	5 sec
3. Pentium vs ENIAC	5 sec
4. Search of the Sum	5 sec
5. Fair Sharing	5 sec
6. The Beach	5 sec
7. Sequence	5 sec
8. Metamorphoses	5 sec
9. Lists and Columns	5 sec

1. Dividing into Groups

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 sec*

Set of natural numbers must be divided into G groups so, that numbers n, $2 \cdot n$, …, $G \cdot n$ correspond to different groups, where n is arbitrary natural number.

Input

The first line of the input file contains two integers M and G, separated by a space, where M is the quantity of numbers in the file ($1 \le M \le 50000$) and G is the quantity of groups ($2 \le G \le 5$). The following M lines contain a single integer n in a line ($1 \le n \le 10^9$).

Output

Write into the output file M lines with numbers of groups for all numbers of the input file. If i-th number of the input file corresponds to k-th group while dividing into G groups, then on i-th line of the output file there should be the number k ($1 \le k \le G$).

Sample of input

```
5 2
1
2
3
4
6
```

Output for the sample input

```
1
2
1
1
2
```

2. The Travel of the Queen.

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 сек*

Once Alice and Bob decided to play chess, but they found out that a lot of chessmen are missing. Then they invented another game. They needed only chess-clock, chessboard and the queen. The rules of the game are quite easy. The first player puts the queen at any field of the chessboard. Then the players move the queen in turn, starting from the second player. If the player puts the queen on the field, where it has already been, he loses. The game can be completed in a draw, only if the queen has already been on every field of the chessboard, but the player may move the queen on the starting field. The queen can move on any number of fields horizontally, vertically or diagonally.

Suppose, that the game is completed in a draw, and all moves of the first player are known. Restore the moves of the second player.

Input

The input file contains one line with the moves of the first player, separated by one space. Verticals are designated by letters from a through h, and horizontals by numerals.

Output

Write into the output file in the same form the moves made by the second player so, that the game is ended in a draw. If there exist a number of variants, then write any one of them.

Sample of input

```
a8 a7 c4 d1 b4 a1 a3 c2 b3 b8 c5 d8 b7 c7 d2 d3 h5 e1 h1 f3 f2 g2 e6 f4 g6 e4 f8 g7 e8 g5 h2 h3
```

Output for the sample input

```
a5 a6 a4 b1 c3 a2 c1 b2 b5 d6 b6 c8 c6 d7 d4 d5 e5 h4 f1 e2 g1 g4 e3 g3 f5 e7 f6 f7 g8 h6 h7 h8
```

3. Pentium vs ENIAC

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 sec*

The calculation of 2000 digits of number π on the computer ENIAC in 1949 took 70 hours (not including programming!). Modern computers (and programmers) can find 2000 digits of number π much faster.

For calculation a series should be used

$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - \ldots$

But this series converges slowly. Much better is series for an arctangent

$\text{arctg}\, x = x - x^3/3 + x^5/5 - x^7/7 + \ldots, |x|<1.$

Combine it with the formula for addition of a tangent

$\text{tg}(a+b) = (\text{tg}\, a + \text{tg}\, b)/(1 - \text{tg}\, a \cdot \text{tg}\, b)$

and select a and b so that $\text{tg}\,(a+b) = 1 = \text{tg}\, \pi/4$. In practice usually use the following formulas

$\pi = 16\ \text{arctg}\ (1/5) - 4\ \text{arctg}\ (1/239).$

$\pi = 32\ \text{arctg}\ (1/10) - 16\ \text{arctg}\ (1/515) - 4\ \text{arctg}\ (1/239).$

$\pi = 12\ \text{arctg}\ (1/4) + 4\ \text{arctg}\ (1/20) + 4\ \text{arctg}\ (1/1985).$

In all formulas it is necessary to calculate $\text{arctg}\ (1/k)$, where $k \geq 2$. Write the program that fulfills this calculation.

Input

The first line of the input file contains one integer k ($2 \leq k \leq 10000$).

Output

Write into the output file a value of $\text{arctg}\ (1/k)$ with accuracy 10^{-2000}.

Sample of input

```
2
```

Output for the sample input (one line)

```
0.463647609000806116214256231461214402028537054286120263810933088720197864165741705300
6002839848878925565298522511908375135058181816250111554715305699441056207193362661648801
0153250275598792580551685388916747823728653879391801251719948401395583818511509502163333
0649387215460973207855555720860146322756524267305218045746400869745058389736389648900026
4868778537801282363312171645781468369009933405288824862445623881190901589497679971970114
4967760016450062530168121256093353041349396630129319242748402931611194920616208441593723
6127316687698168702759318951033397332592903851289254594592246321560978363800953749932094
9486073394918643251602748279304503733177255465049960867577062275441628502227372371197447
7336697731851069401381126995777925627482566009621167267481152728272252072259726842157101
9587756209170155776870986654266890344935180547289005370783812421285479430302436784526466
9937683808877190412767311593748061628833032028804465239589618924130515270876726439400070
4439235424425691226977711518927717226446341501457164858901254102646277708194745101871218
4820604772739804643995734922413959126815178050223278635514347337079482401251533058001510
9270674591998921707936339487128099705106434535905756066721321179928796810834788482208814
2360467482775500514191422286419991496650851144084719924136084016015527819437134308245579
7430238250890332380343406851847378786512745224357245815283117590020057033343634072258264
8575409343821252189315998634896079848398950569509327230867566007767526527325937527348914
8415686631500665360279054807579044426860451551444817706217292797414080221775825566238236
6921134153548554767355969874040365137652336060455236185813448715006856094114280464829259
9332129615556169392765012713663663644382665370304068180900070642920215430207040940370202
4169509775647904851545073314861212196353193833893702959673994983774315170452766866326065
9549613673180890291569450774137681640705477520727671692403801676155508214257282171011716
4287897755349658231508880156622408053339948132487758071305256952539999844
8
```

4. Search of the Sum

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 sec*

The sequence of M integers is given. Find here the longest continuous subsequence with the sum of terms equal to given S.

Input

The first line of the input file contains two integers M and S, separated by one space, where M is the quantity of numbers in the sequence ($2 \le M \le 30000$) and S is the value for the sum of terms of a subsequence ($0 \le S \le 1000$). The following M lines contain a single integer n in a line ($-10000 \le n \le 10000$).

Output

On the first line of the output file write two integers F and L, separated by one space, where F is the index of the first term of the retrieved subsequence (indexing starts from 1), and L is the length of the retrieved subsequence. If there are some equally long subsequences, indicate the first of them in the source sequence. If a subsequence with the given sum of terms is missing, write two zeros, separated by one space.

Sample of input

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

Output for the sample input

```
1 2
```

5. Fair Sharing

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 sec*

Gnomes snatched away the sweets from a vase, which Snow-white had place on the round table.

“Stop! You haven’t left any sweets for me!” – Snow-white exclaimed. – “We need to share the sweets equally.”

Snow-white counted the sweets; there were 15 sweets.

“Fifteen cannot be divided on eight. Let’s do it like this. Put your sweets into bowls.” – After it had been done, Snow-white continued. – “Let's enumerate bowls clockwise from 1 up to 8. My bowl will be under number 1. I will turn away and call the numbers from 1 to 8. Somebody takes all the sweets from the called bowl and distribute one in each bowl clockwise, starting from the next one after the called. I think, after several such operations the sweets will be distributed on cups in random.”

The gnomes agreed, that this sharing would be fairer, as the results will depend on case, but not on “the length of raking arms” (Grumbler said so, who had got only one sweet). But as a result Grumbler got nothing, because Snow-white was offended with him for grumbling about the quality of yesterday’s dinner. Snow-white differed from her stepmother not only in beauty, but also in wit (at least, it did not come into her mind to talk to a mirror), therefore Snow-white easily invented, how to achieve more fair distribution (from her point of view) from initial distribution of sweets among bowls.

Write the program, which determines according to initial and finite distribution of 15 sweets among 8 bowls a sequence of numbers, which needs to be pronounced by Snow-white.

Input

The first line of the input file contains 8 integers (in the range from 0 to 15 inclusively, the sum is 15), separated by spaces; this is initial distribution of sweets among bowls (clockwise, starting from a bowl #1). Second line contains 8 integers (in the range from 0 to 15 inclusively, the sum is 15), separated by spaces; this is required finite distribution of sweets among bowls (clockwise, starting from a bowl #1).

Output

On the first line of the output file write one integer N representing an amount of called numbers. The following N lines of the output file with numbers from 1 up to 8 representing a sequence of called numbers. Any variant, even not necessarily the shortest, may be pointed.

Sample of input

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

Output for the sample input

```
7
2
4
6
5
8
7
3
```

6. The Beach

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 sec*

Bill has bought some islands at the ocean in the gross and has decided to sell them off piecemeal. But the buyers are interested only in the sites located on the coast of the ocean.

Bill divided the map of islands into square cells and represented each cell by one character. Bill used the character '.' (dot) for the denotation of water and character '*' (asterisk) for the land. Each character '*' corresponds to one site for sale. There are no other characters on the map.

Write the program, which will help Bill to determine an amount of sites having common boundary with the ocean according to this map. The sites, adjoining only to a lake or a pool, are not taken into account.

Input

The first line of the input file contains two integers N and M, separated by one space, where N and M are the sizes of the map ($3 \le N \le 200$, $3 \le M \le 200$). The following N lines contain M characters with each line representing a map of the bought islands. NOTE: The islands do not adjoin to the edges of the map.

Output

Write in the output file an amount of sites located on the coast of the ocean.

Sample of input

```
8 10
..........
..***.....
.*****....
.**..*....
..****....
........*.
......***.
..........
```

Output for the sample input

```
15
```

7. Sequence

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 sec*

Let's take any natural number K and generate an infinite sequence S according to the following rules.

1. S_i is i, where $1 \le i \le K$
2. S_i is the least natural number so, that gcd (S_i, S_{i-1}) $\ge K$ and $S_i \ne S_j$, where $1 \le j < i$ and $i > K$ and gcd mean "greatest common divisor".

For example, for K=3 first twenty terms are 1, 2, 3, 6, 9, 12, 4, 8, 16, 20, 5, 10, 15, 18, 21, 7, 14, 28, 24, 27.

Sooner or later in this sequence there will be all natural numbers. Write the program, which calculates the least natural number that is missing among first N terms, the greatest number among first N terms, and N-th term.

Input

The first line of the input file contains two integers K and N ($1 \le K \le 100$, $1 \le N \le 10000$), separated by one space. First N terms of the sequence S do not exceed 30000 for given K.

Output

Write into the output file the one line containing three integers, separated by spaces. They are the least natural number that is missing among first N terms of the sequence S, the greatest number among first N terms of the sequence, and N-th term.

Sample of input

```
3 20
```

Output for the sample input

```
11 28 27
```

8. Metamorphoses

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 sec*

Take two words of equal length, for example, SEAT and TALE, write them one under another and make the following transformations. At first, replace all appearances of the first character of the second word in the second word with the first character of the first word, and all appearances of the first character of the first word in the first word with the first character of the second word. Similarly, all appearances of the second character of the second word in the second word with the second character of the first word, and all appearances of the second character of the first word with the first word on the second character of the second word. The same must be done for the third pair of characters of the both words, and so on for all characters from left to right.

```
SEAT   TEAT   TAAT   TLLT   ELLE
    →      →      →      →
TALE   SALE   SELE   SEAE   STAT
```

Write the program, which calculates outcome of transformations for the pair of the given words.

Input

Input file contains two lines. Each line contains one word, the length of which is not more than 100 characters consisting of uppercase letters from A through Z. Both words have equal length.

Output

Write into the output file two lines containing the outcome of transformations. On the first line write the transformed first word, and on the second line write the transformed second word.

Sample of input

```
SEAT
TALE
```

Output for the sample input

```
ELLE
STAT
```

9. Lists and Columns

Input file	*INPUT.TXT*
Output file	*OUTPUT.TXT*
Time limit	*5 sec*

Write the program, which will transform lists of words into table columns. The first column should contain words of the first list, second column should contain words of the second list, and so on. Words should be written without skips in same order as in the lists. The width of a column is equal to the length of the longest word of this column. Columns should be separated by one space.

Input

The input file contains N lines ($1 \le N \le 100$) of length up to 200 characters. Each line contains the nonempty list of words consisting of letters. Words are separated by commas. The lines do not contain spaces. The new-line character ends each line.

Output

Write into the output file the table of N columns containing all words from lists. No spaces are allowed at the end of lines.

Sample of input

```
Bill,Jo,Robert,Sally
Ann,Sam,Carolyn
fire,grain,cat,column,store
```

Output for the sample input

```
Bill   Ann     fire
Jo     Sam     grain
Robert Carolyn cat
Sally          column
               store
```