A. Divisible by Seven

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You have number a, whose decimal representation quite luckily contains digits 1, 6, 8, 9. Rearrange the digits in its decimal representation so that the resulting number will be divisible by 7.

Number a doesn't contain any leading zeroes and contains digits 1, 6, 8, 9 (it also can contain another digits). The resulting number also mustn't contain any leading zeroes.

Input

The first line contains positive integer a in the decimal record. It is guaranteed that the record of number a contains digits: 1, 6, 8, 9. Number a doesn't contain any leading zeroes. The decimal representation of number a contains at least 4 and at most 10^6 characters.

Output

Print a number in the decimal notation without leading zeroes — the result of the permutation.

If it is impossible to rearrange the digits of the number a in the required manner, print 0.

Sample test(s)

input

```
1689
```

output

```
1869
```

input

```
18906
```

output

```
18690
```

B. Maximum Submatrix 2

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given a matrix consisting of digits zero and one, its size is $n \times m$. You are allowed to rearrange its rows. What is the maximum area of the submatrix that only consists of ones and can be obtained in the given problem by the described operations?

Let's assume that the rows of matrix a are numbered from 1 to n from top to bottom and the columns are numbered from 1 to m from left to right. A matrix cell on the intersection of the i-th row and the j-th column can be represented as (i, j). Formally, a submatrix of matrix a is a group of four integers d, u, l, r $(1 \le d \le u \le n;\ 1 \le l \le r \le m)$. We will assume that the submatrix contains cells (i, j) $(d \le i \le u;\ l \le j \le r)$. The area of the submatrix is the number of cells it contains.

Input

The first line contains two integers n and m $(1 \le n, m \le 5000)$. Next n lines contain m characters each — matrix a. Matrix a only contains characters: "`0`" and "`1`". Note that the elements of the matrix follow without any spaces in the lines.

Output

Print a single integer — the area of the maximum obtained submatrix. If we cannot obtain a matrix of numbers one, print 0.

Sample test(s)

input

```
1 1
1
```

output

```
1
```

input

```
2 2
10
11
```

output

```
2
```

input

```
4 3
100
011
000
101
```

output

```
2
```

C. Circling Round Treasures

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a map as a rectangle table. Each cell of the table is either an obstacle, or a treasure with a certain price, or a bomb, or an empty cell. Your initial position is also given to you.

You can go from one cell of the map to a side-adjacent one. At that, you are not allowed to go beyond the borders of the map, enter the cells with treasures, obstacles and bombs. To pick the treasures, you need to build a closed path (starting and ending in the starting cell). The closed path mustn't contain any cells with bombs inside. Let's assume that the sum of the treasures' values that are located inside the closed path equals v, and besides, you've made k single moves (from one cell to another) while you were going through the path, then such path brings you the profit of $v - k$ rubles.

Your task is to build a closed path that doesn't contain any bombs and brings maximum profit.

Note that the path can have self-intersections. In order to determine if a cell lies inside a path or not, use the following algorithm:

1. Assume that the table cells are points on the plane (the table cell on the intersection of the i-th column and the j-th row is point (i, j)). And the given path is a closed polyline that goes through these points.
2. You need to find out if the point p of the table that is not crossed by the polyline lies inside the polyline.
3. Let's draw a ray that starts from point p and does not intersect other points of the table (such ray must exist).
4. Let's count the number of segments of the polyline that intersect the painted ray. If this number is odd, we assume that point p (and consequently, the table cell) lie inside the polyline (path). Otherwise, we assume that it lies outside.

Input

The first line contains two integers n and m $(1 \le n, m \le 20)$ — the sizes of the table. Next n lines each contains m characters — the description of the table. The description means the following:

- character "`B`" is a cell with a bomb;
- character "`S`" is the starting cell, you can assume that it's empty;
- digit c (`1`-`8`) is treasure with index c;
- character "`.`" is an empty cell;
- character "`#`" is an obstacle.

Assume that the map has t treasures. Next t lines contain the prices of the treasures. The i-th line contains the price of the treasure with index i, v_i $(-200 \le v_i \le 200)$. It is guaranteed that the treasures are numbered from 1 to t. It is guaranteed that the map has not more than 8 objects in total. Objects are bombs and treasures. It is guaranteed that the map has exactly one character "`S`".

Output

Print a single integer — the maximum possible profit you can get.

Sample test(s)

input

```
4 4
....
.S1.
....
....
10
```

output

```
2
```

input

```
7 7
.......
.1###2.
.#...#.
.#.B.#.
.3...4.
..##...
......S
100
100
100
100
```

output

```
364
```

input

```
7 8
........
```

```
........
....1B..
.S......
....2...
3.......
........
100
-100
100
```

output

```
0
```

input

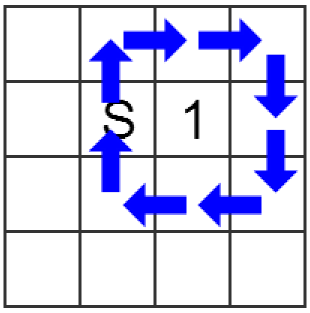
```
1 1
S
```

output

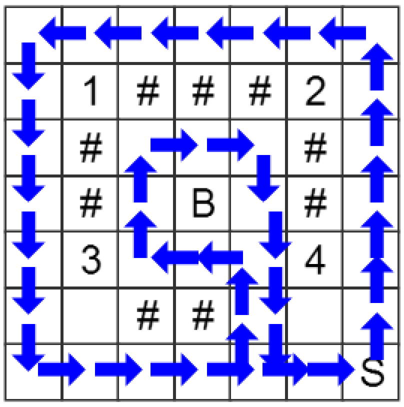
```
0
```

Note

In the first example the answer will look as follows.



In the second example the answer will look as follows.



In the third example you cannot get profit.

In the fourth example you cannot get profit as you cannot construct a closed path with more than one cell.

D. Tree and Queries

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a rooted tree consisting of n vertices. Each vertex of the tree has some color. We will assume that the tree vertices are numbered by integers from 1 to n. Then we represent the color of vertex v as c_v. The tree root is a vertex with number 1.

In this problem you need to answer to m queries. Each query is described by two integers v_j, k_j. The answer to query v_j, k_j is the number of such colors of vertices x, that the subtree of vertex v_j contains at least k_j vertices of color x.

You can find the definition of a rooted tree by the following link: `http://en.wikipedia.org/wiki/Tree_(graph_theory)`.

Input

The first line contains two integers n and m $(2 \le n \le 10^5;\ 1 \le m \le 10^5)$. The next line contains a sequence of integers $c_1, c_2, ..., c_n$ $(1 \le c_i \le 10^5)$. The next n - 1 lines contain the edges of the tree. The i-th line contains the numbers a_i, b_i $(1 \le a_i, b_i \le n;\ a_i \ne b_i)$ — the vertices connected by an edge of the tree.

Next m lines contain the queries. The j-th line contains two integers v_j, k_j $(1 \le v_j \le n;\ 1 \le k_j \le 10^5)$.

Output

Print m integers — the answers to the queries in the order the queries appear in the input.

Sample test(s)

input

```
8 5
1 2 2 3 3 2 3 3
1 2
1 5
2 3
2 4
5 6
5 7
5 8
1 2
1 3
1 4
2 3
5 3
```

output

```
2
2
1
0
1
```

input

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

output

```
4
```

Note

A subtree of vertex v in a rooted tree with root r is a set of vertices $\{u : dist(r, v) + dist(v, u) = dist(r, u)\}$. Where $dist(x, y)$ is the length (in edges) of the shortest path between vertices x and y.

E. Red and Black Tree

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a weighted tree, consisting of n vertices. Each vertex is either painted black or is painted red. A red and black tree is called *beautiful*, if for any its vertex we can find a black vertex at distance at most x.

The distance between two nodes is the shortest path between them.

You have a red and black tree. Your task is to make it beautiful in the minimum number of color swap operations. In one color swap operation, you can choose two vertices of different colors and paint each of them the other color. In other words, if you choose a red vertex p and a black vertex q, then in one operation you are allowed to paint p black and paint q red.

Print the minimum number of required actions.

Input

The first line contains two integers n and x $(2 \le n \le 500;\ 1 \le x \le 10^9)$. The next line contains n integers, each of them is either a zero or one. If the i-th number equals 1, then vertex i of the tree is black, otherwise vertex i is red. Next $n - 1$ lines contain the tree edges. The j-th line contains integers u_j v_j w_j $(1 \le u_j, v_j \le n;\ u_j \ne v_j;\ 1 \le w_j \le 10^9)$ which means that the tree has an edge of weight w_j between vertices v_j and u_j.

Assume that the tree vertices are numbered from 1 to n.

Output

Print a single integer — the minimum number of required swap operations.

If it is impossible to get a beautiful tree at any number of operations, print -1.

Sample test(s)

input

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

output

```
1
```

input

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

output

```
-1
```