E. Remembering Strings

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have multiset of n strings of the same length, consisting of lowercase English letters. We will say that those strings are easy to remember if for each string there is some position i and some letter c of the English alphabet, such that this string is the only string in the multiset that has letter c in position i.

For example, a multiset of strings {"abc", "aba", "adc", "ada"} are not easy to remember. And multiset {"abc", "ada", "ssa"} is easy to remember because:

- the first string is the only string that has character c in position 3;
- the second string is the only string that has character d in position 2;
- the third string is the only string that has character s in position 2.

You want to change your multiset a little so that it is easy to remember. For a_{ij} coins, you can change character in the j-th position of the i-th string into any other lowercase letter of the English alphabet. Find what is the minimum sum you should pay in order to make the multiset of strings easy to remember.

Input

The first line contains two integers n, m $(1 \le n, m \le 20)$ — the number of strings in the multiset and the length of the strings respectively. Next n lines contain the strings of the multiset, consisting only of lowercase English letters, each string's length is m.

Next n lines contain m integers each, the i-th of them contains integers $a_{i1}, a_{i2}, ..., a_{im}$ $(0 \le a_{ij} \le 10^6)$.

Output

Print a single number — the answer to the problem.

Examples

input

```
4 5
abcde
abcde
abcde
abcde
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
```

output

```
3
```

input

```
4 3
abc
aba
adc
ada
10 10 10
10 1 10
10 10 10
10 1 10
```

output

```
2
```

input

```
3 3
abc
ada
ssa
1 1 1
1 1 1
1 1 1
```

output

```
0
```