G - Pipes

Input: `standard input`
Output: `standard output`

The construction of office buildings has become a very standardized task. Pre-fabricated modules are combined according to the customers needs, shipped from a faraway factory, and assembled on the construction site. However, there are still some tasks that require careful planning, one example being the routing of pipes for the heating system.

A modern office building ismade up of squaremodules, one on each floor being a service module from which (among other things) hot water is pumped out to the other modules through the heating pipes. Each module (including the service module) will have heating pipes connecting it to exactly two of its two to four neighboring modules. Thus, the pipes have to run in a circuit, from the service module, visiting each module exactly once, before finally returning to the service module. Due to different properties of the modules, having pipes connecting a pair of adjacent modules comes at different costs. For example, some modules are separated by thick walls, increasing the cost of laying pipes. Your task is to, given a description of a floor of an office building, decide the cheapest way to route the heating pipes.

Input

The first line of input contains a single integer, stating the number of floors to handle. Then follow n floor descriptions, each beginning on a new line with two integers, $2 \leq r \leq 10$ and $2 \leq c \leq 10$, defining the size of the floor - r-by-c modules. Beginning on the next line follows a floor description in ASCII format, in total $2r + 1$ rows, each with $2c + 2$ characters, including the final newline. All floors are perfectly rectangular, and will always have an even number of modules. All interior walls are represented by numeric characters, '0' to '9', indicating the cost of routing pipes through the wall (see sample input).

Output

For each test case, output a single line with the cost of the cheapest route.

Sample Input

```
3
4 3
#######
# 2 3 #
#1#9#1#
# 2 3 #
#1#7#1#
# 5 3 #
#1#9#1#
# 2 3 #
#######
```

```
4 4
#########
# 2 3 3 #
#1#9#1#4#
# 2 3 6 #
#1#7#1#5#
# 5 3 1 #
#1#9#1#7#
# 2 3 0 #
#########
2 2
#####
# 1 #
#2#3#
# 4 #
#####
```

Sample Output

```
28
45
10
```