Problem 1: Bessie Slows Down

Bessie the cow is competing in a cross-country skiing event at the winter Moolympic games. She starts out at a speed of 1 meter per second. However, as she becomes more tired over time, she begins to slow down. Each time Bessie slows down, her speed decreases: she moves at 1/2 meter per second after slowing down once, then 1/3 meter per second after slowing down twice, and so on.

You are told when and where Bessie slows down, in terms of a series of events. An event like this:

T 17

means that Bessie slows down at a specific time -- here, 17 seconds into the race. An event like this:

D 10

means that Bessie slows down at a specific distance from the start -- in this case, 10 meters.

Given a list of N such events (1 <= N <= 10,000), please compute the amount of time, in seconds, for Bessie to travel an entire kilometer. Round your answer to the nearest integer second (0.5 rounds up to 1).

INPUT FORMAT:

* Line 1: The value of N.

* Lines 2..1+N: Each line is of the form "T x" or "D x", indicating a time event or a distance event. In both cases, x is an integer that is guaranteed to place the event before Bessie reaches one kilometer of total distance. It is possible for multiple events to occur simultaneously, causing Bessie to slow down quite a bit all at once. Events may not be listed in order.

SAMPLE INPUT (file slowdown.in):

2

T 30
D 10

INPUT DETAILS:

Bessie slows down at time t = 30 and at distance d = 10.

OUTPUT FORMAT:

* Line 1: The total time required for Bessie to travel 1 kilometer.

SAMPLE OUTPUT (file slowdown.out):

2970

OUTPUT DETAILS:

Bessie travels the first 10 meters at 1 meter/second, taking 10 seconds. She then slows down to 1/2 meter/second, taking 20 seconds to travel the next 10 meters. She then reaches the 30 second mark, where she slows down again to 1/3 meter/second. The remaining 980 meters therefore take her 980 * 3 = 2940 seconds. The total time is therefore 10 + 20 + 2940 = 2970.

Problem 2: Balanced Teams

A total of 12 of Farmer John's cows are attending the winter Moolympic games this year, each with an integer skill level between 1 and 1,000,000.

Farmer John wants to divide them into 4 teams of 3, so that the teams come out reasonably "balanced" in terms of total skill (the skill level of a team is just the sum of the skill levels of the cows on the team). Specifically, he wants to minimize S - s, where S and s are the maximum and minimum skill levels of the teams. This ensures that the variation between the most-skilled and least-skilled teams is as small as possible.

Please help Farmer John determine the minimum possible value of S - s.

INPUT FORMAT:

* Lines 1..12: Each line contains the skill level of a single cow.

SAMPLE INPUT (file bteams.in):

```
1
2
3
4
5
6
7
8
9
10
11
12
```

OUTPUT FORMAT:

* Line 1: The minimum possible value of S - s.

SAMPLE OUTPUT (file bteams.out):

```
1
```

OUTPUT DETAILS:

One possible solution is to divide the teams as follows: (12,1,7), (9,8,3), (10,5,4), and (11,2,6). The first two have skill 20 and the second two have skill 19.

Problem 3: Recording the Moolympics

Being a fan of all cold-weather sports (especially those involving cows), Farmer John wants to record as much of the upcoming winter Moolympics as possible.

The television schedule for the Moolympics consists of N different programs (1 <= N <= 150), each with a designated starting time and ending time. FJ has a dual-tuner recorder that can record two programs simultaneously. Please help him determine the maximum number of programs he can record in total.

INPUT FORMAT:

* Line 1: The integer N.

* Lines 2..1+N: Each line contains the start and end time of a single program (integers in the range 0..1,000,000,000).

SAMPLE INPUT (file recording.in):

```
6
0 3
6 7
3 10
1 5
2 8
1 9
```

INPUT DETAILS:

The Moolympics broadcast consists of 6 programs. The first runs from time 0 to time 3, and so on.

OUTPUT FORMAT:

* Line 1: The maximum number of programs FJ can record.

SAMPLE OUTPUT (file recording.out):

```
4
```

OUTPUT DETAILS:

FJ can record at most 4 programs. For example, he can record programs 1 and 3 back-to-back on the first tuner, and programs 2 and 4 on the second tuner.

Problem 4: Building a Ski Course

Farmer John is helping to turn his large field into a ski course for the upcoming winter Moolympics. The field has dimensions M x N (1 <= M,N <= 100), and its intended final composition is described by an M x N grid of characters like this:

RSRSSS
RSRSSS
RSRSSS

Each character describes how the snow in a unit square of the field should be groomed: either 'R' for 'rough' or 'S' for 'smooth' (the Moolympics organizers think that a course is more interesting if it has a mixture of rough and smooth patches).

To build the desired course, Farmer John plans to modify his tractor so that it can stamp any B x B patch of the field (B <= M, B <= N) with either entirely smooth snow or entirely rough snow. Since it takes a long time to reset the tractor between each of these stamps, FJ wants to make B as large as possible. With B = 1, he can clearly create the desired ski course by stamping each individual square with either R or S, as intended. However, for larger values of B, it may no longer be possible to create the desired course design. Every unit square of the course must at some point be stamped by FJ's tractor; it cannot be left in its default state.

Please help FJ determine the largest possible value of B he can successfully use.

INPUT FORMAT:

* Line 1: Two space-separated integers M and N.

* Lines 2..M+1: M lines of exactly N characters (each R or S), describing the desired ski course design.

SAMPLE INPUT (file skicourse.in):

3 6
RSRSSS
RSRSSS

RSRSSS

OUTPUT FORMAT:

* Line 1: The maximum value of B Farmer John can use to create the
  desired course pattern.

SAMPLE OUTPUT (file skicourse.out):

3

OUTPUT DETAILS:

FJ can stamp a rough patch spanning columns 1-3, followed by a smooth patch spanning columns 2-4, then a rough patch spanning columns 3-5, and finally a smooth patch spanning columns 4-6.

Problem 5: Ski Course Rating

The cross-country skiing course at the winter Moolympics is described by an M x N grid of elevations (1 <= M,N <= 500), each elevation being in the range 0 .. 1,000,000,000.

Some of the cells in this grid are designated as starting points for the course. The organizers of the Moolympics want to assign a difficulty rating to each starting point. The difficulty level of a starting point P should be the minimum possible value of D such that a cow can successfully reach at least T total cells of the grid (1 <= T <= MN), if she starts at P and can only move from cell to adjacent cell if the absolute difference in elevation between the cells is at most D. Two cells are adjacent if one is directly north, south, east, or west of the other.

Please help the organizers compute the difficulty rating for each starting point.

INPUT FORMAT:

* Line 1: The integers M, N, and T.

* Lines 2..1+M: Each of these M lines contains N integer elevations.

* Lines 2+M..1+2M: Each of these M lines contains N values that are either 0 or 1, with 1 indicating a cell that is a starting point.

SAMPLE INPUT (file skilevel.in):

```
3 5 10
20 21 18 99 5
19 22 20 16 17
18 17 40 60 80
1 0 0 0 0
0 0 0 0 0
0 0 0 0 1
```

INPUT DETAILS:

The ski course is described by a 3 x 5 grid of elevations. The upper-left

and lower-right cells are designated as starting points. From each starting point, we must be able to reach at least 10 cells.

OUTPUT FORMAT:

* Line 1: The sum of difficulty ratings of all starting points (note that this may not fit into a 32-bit integer, even though individual difficulty ratings will).

SAMPLE OUTPUT (file skilevel.out):

24

OUTPUT DETAILS:

The difficulty rating of the upper-left starting point is 4, and for the lower-right it is 20.