2008 Canadian Computing Competition
Day 2, Question 1

Problem D – Herding

Input: from standard input
Output: to standard output
Source file: `herding.{c, cpp, pas}`

Oh no! A number of stray cats have been let loose in the city, and as the City Cat Catcher, you have been assigned the vital task of retrieving all of the cats. This is an ideal opportunity to test your latest invention, a cat trap which is guaranteed to retrieve every cat which walks into a square-shaped subsection of the city.

Fortunately, you have the assistance of one of the world's foremost cat psychologists, who has the amazing ability of predicting, given a square subsection of the city, exactly which of the four cardinal directions (north, east, south or west) the cat will head. While this information is handy, you still don't know where all the cats currently are.

In order to prove the cost-effectiveness of your method to the City it would, of course, be important to minimize the number of traps used.

Input description

The input will begin with a line consisting of two numbers n and m, separated by a space ($1 \leq n, m \leq 1000$). The city will be an $n \times m$ grid of square subsections. The next n lines will each consist of a string of length m, consisting of the letters 'N', 'E', 'S', or 'W', representing north, east, south and west, respectively. (The first character of the first line will be the northwesternmost point.) The direction in the square is the direction which cats will head if they are in that square. The cat psychologist assures you that cats have no interest in leaving the city.

Output description

Output the minimum number of traps needed.

Sample input

```
3 4
SWWW
SEWN
EEEN
```

Output for sample input

Grading

You may assume that 40% of test cases will have $1 \leq n, m \leq 4$. All test cases will have $1 \leq n, m \leq 1000$. You solution must use at most 512MB of memory and run in at most 3 seconds.

2008 Canadian Computing Competition
Day 2, Question 2

Problem E – Candy

Input: from standard input
Output: to standard output
Source file: `candy.{c, cpp, pas}`

You and a friend have a big bag of candy. You want to keep slim and trim, and so you would like to equalize the candy which you are sharing with your friend in terms of calorie count. That is, your task is to divide the candies into two groups such that the number of calories in each group is as close together as possible.

Input description

The first line of input contains the number of different kinds of candy you have in your bag of candy N ($1 \leq N \leq 100$). On the following N lines, there are pairs of numbers describing each type of candy. The candy description is of the form k_i c_i where k_i is the number of that particular type of candy contained in the bag and c_i is the calorie count for each piece of that type of candy. You may assume that $1 \leq k_i \leq 500$ and $1 \leq c_i \leq 200$.

Output description

Your output is one integer which is the minimum difference of calories between friends

Sample input

```
4
3 5
3 3
1 2
3 100
```

Output for sample input

```
74
```

Explanation for sample output

Your friend takes two of the 100-calorie candies, for a total of 200 calories. You keep the remaining candies, which have 126 calories.

Grading

You may assume that 50% of the test cases will have at $1 \leq N, k_i, c_i \leq 100$. All test cases will have $1 \leq N \leq 100$, $1 \leq k_i \leq 500$ and $1 \leq c_i \leq 200$. You solution must use at most 512MB of memory and run in at most 6 seconds.

2008 Canadian Computing Competition
Day 2, Question 3

Problem F – Landing

Input: from standard input
Output: to standard output
Source file: `landing.{c, cpp, pas}`

Keep watching the skies! Alien spacecraft are due to land any day now to share all of their advanced programming secrets with us.

In preparation for this day, you've been tasked with preparing a landing pad for our visitors in a given field. Unfortunately, due to environmental considerations, you will not be permitted to remove any of the trees which currently exist on the field. These trees are of immense scientific interest, since they have zero radius and only grow at points with integer co-ordinates. However, this could be a blessing in disguise. For security reasons, the landing pad must be in contact with at least three trees. Security cameras will be placed at the tops of these trees.

Alien spacecraft are perfectly circular craft of various sizes, so the landing pad will also be circular. Since it would be polite to warn potential visitors ahead of time if their spacecraft is too large for our landing pad, you must now determine the size of the largest circular region that we can place on the field which contacts at least three trees, but does not contain any trees within.

Input description

The first line of input will consist of the number n of trees ($3 \leq n \leq 100000$). The next n lines will each consist of a pair of integers x and y ($-10000 \leq x, y \leq 10000$), separated by a space, giving the co-ordinates of a tree. You may assume that no two trees are at the same co-ordinate.

Output description

Output the radius of the largest possible landing pad. If the correct answer is R, you should output any number a such that

$$\frac{R}{1 + 10^{-4}} < a < R(1 + 10^{-4})$$

The above calculation is used to define an acceptable range or tolerance for the answer you find. You may also assume that $R < 10^9$. You may assume there exists at least one landing pad.

Sample input

```
4
1 1
1 -1
-1 -1
-1 1
```

Output for sample input

```
1.41421356
```

Grading

You may assume that 25% of the test cases will have at most 50 trees. You may assume that 95% of the test cases will have at most 1000 trees. All test cases will have at most 100000 trees. You solution must use at most 512MB of memory and run in at most 5 seconds.