The Gold Rush

Charlie has found a treasure map of Klondike with N points marked as places in which a certain amount of gold is hidden. Unfortunately, one of his so called friends (Big Jim) steals the map and heads for the gold. Hopefully as we are in technology area, Charlie had taken a picture of that map with his smart phone. Now as he knows Big Jim is also after the gold, he wants to get the maximum amount of gold as soon as possible.

Assume Charlie has sorted the N points ((x, y)-coordinates) in a list $POINTS[1...N]$ based on some criteria and he wants to start from point 1 and end in point N and the sequence of points he will visit is going to be increasing in index (i.e. if he is in point i he can move to point j if and only if $i < j$).

If he visits point i, he will get $G[i]$ units of gold which are hidden in that point. Also if he travels the total distance of d (sum of euclidian distances between visited points), he needs to spend d units of gold (note that d can be a real number).

Design and implement an algorithm to help Charlie determine the maximum amount of gold he can make (i.e., the total hidden units of gold he collects minus the units of gold he spends), based on the above assumptions.

Input Format

First line contain integer N. Next N lines contains three integers each, i-th line contains coordinates of i-th points (X_i, Y_i) which comes in $POINTS[i]$ followed by $G[i]$.

Output Format

Output the floor of the maximum units of gold Charlie can get under the given assumptions.

Sample Input 0

```
3
0 0 1
3 1 1
6 0 9
```

Sample Output 0

```
4.675445
```

Explanation 0

If Charlie moves from (0,0) to (3,1) and then (6,0) he will get 11-2*sqrt(10) = 4.6754446 units of gold maximum, and you should report number rounded to 6 digits after floating point which is 4.675445.