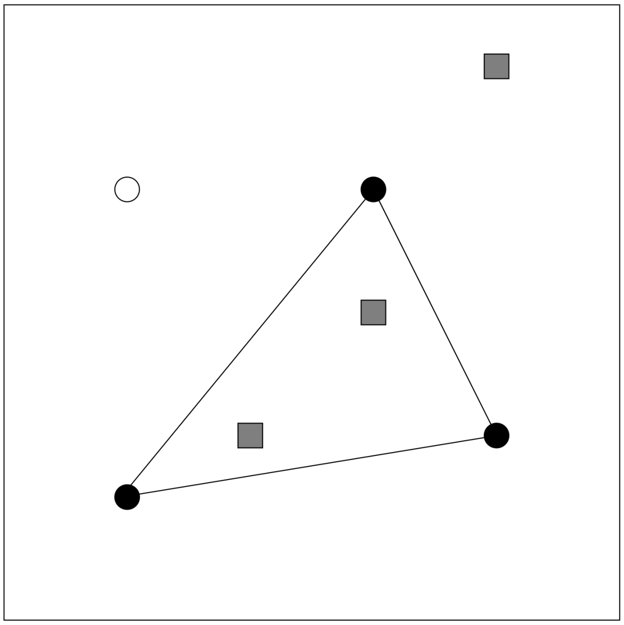
IOI Team Training 2013 – Online Test 9, 9–10 June, 2013

Fence

One morning, fruit farmer Fred visits his apple trees and notices that one of them was cut overnight. This means a loss of $111—the money he can make from the apples of a tree on average. In order to prevent further losses, he decides to erect a fence on his plantation.

The fence consists of posts connected by wire. The fence posts can only be placed at a given set of pre-drilled holes. While Fred can get wire for free, he needs to buy the fence posts for $20 each. So it might not always be worth or even possible to fence in all of his trees.

The plantation is square and $1000 \times 1000m^2$ large. Seen from the sky, the lower left corner has coordinates $(0, 0)$ and the upper right corner has coordinates $(1000, 1000)$.



In this example there are four pre-drilled holes (circles) and three trees (squares). It is optimal to buy three fence posts and put them into selected holes (filled circles), to connect them by wire (lines), and to leave the upper left hole empty. The cost of erecting the fence is $3 \cdot 20 + 1 \cdot 111 = \$171$ since three posts were bought and one tree could not be fenced in (which means a loss of that tree's harvest).

Program and Grader

Implement one function in the file `fence.cpp`

```
int cost(int N, int M, int* xH, int* yH, int* xT, int* yT);
```

that returns the minimum cost you will incur by setting up the fence and loss of your trees, where `N` is the number of pre-drilled holes, `M` is the number of trees, `xH` and `yH` are N-element arrays containing the x-coordinates and y-coordinates of the holes, respectively, and `xT` and `yT` are M-element arrays containing the x-coordinates and y-coordinates of the trees, respectively.

Input and Output

The program `grader_fence.cpp` takes input in the following format:

- The first line has two integers: N and M.
- The next N lines contain two integers each, `xH[i]` and `yH[i]`.
- The next M lines contain two integers each, `xT[i]` and `yT[i]`.

`grader_fence.cpp` then calls your function `cost()` and outputs the integer that your function returns.

Test data

No three positions will be collinear. No two positions will coincide. For all positions, $0 \leq x, y \leq 1000$, and $1 \leq M \leq 100$.

- Subtask 1 (25 marks) : $3 \leq N \leq 15$
- Subtask 2 (75 marks) : $3 \leq N \leq 100$

Sample input

```
43
800 300
200 200
200 700
600 700
400 300
600 500
800 900
```

Sample output

```
171
```

Limits

- *Memory limit* : 128 MB
- *Time limit* : 4s