

До окончания тестирования осталось 6 дней 8 часов 19 минут

Задача 2. Финансовая фантазия фанатичного фермера

Условие задачи

Ограничение времени, с	1
Ограничение памяти, МБ	64
Общее число попыток отправки	15

Фермер Василий выбирает землю для покупки. Предмет торгов – прямоугольное поле шириной n и высотой m , которое состоит из участков, где **1** - плодородный участок, а **0** – неплодородный. Василий может либо купить регион поля любого размера, либо отказаться от покупки, если доступных для покупки регионов нет.

Условия покупки следующие:

- Регион – это прямоугольник, ограничивающий соприкасающиеся участки плодородной почвы
- Участки "соприкасаются" если они соседние друг для друга – сверху, снизу, справа, слева и по диагонали

```
1 0 1
0 1 1
1 0 1
0 0 0
0 1 0
```

На примере выше соприкасаются все участки, кроме нижнего, то есть регионов здесь 2, один площадью 9, другой площадью 1

- Регионы могут пересекаться между собой:

```
1 1 1 1 1
1 0 0 0 1
1 0 1 0 1
```

Здесь тоже два региона, один площадью 15 (все поле), другой площадью 1

- Минимальное количество плодородных участков в регионе для покупки – 2
- Покупатель платит только за общую площадь купленного региона

Василий берет кредит на покупку, поэтому хочет потратить деньги как можно оптимальнее – купить тот регион, в котором будет максимальное соотношение плодородной земли к общей площади региона. Если есть несколько регионов с одинаковой «эффективностью», то Василий хочет купить больший из них по площади.

Нужно определить площадь региона, который стоит купить фермеру

Входные данные (поступают в стандартный поток ввода)

Первая строка – целые числа n, m через пробел ($2 \leq n \leq 100, 2 \leq m \leq 100$)

Далее m строк, в каждой из которых по n цифр 0 или 1, разделенных пробелами

Все входные данные наших тестов всегда соблюдают указанные параметры, дополнительные проверки не требуются

Выходные данные (ожидаются в стандартном потоке вывода)

Одно целое число, площадь наилучшего региона, или 0, в случае отказа от покупки

Пример 1

Ввод:

```
5 4
0 1 1 0 0
1 1 1 0 1
1 1 0 0 1
```

0 0 0 1 0

Вывод:

9

На этом поле доступны для покупки:

Первый регион для покупки

Левый верхний угол с координатами [0, 0]

Правый нижний угол с координатами [2, 2]

Его площадь 9, а плодородных участков на нем 7.

Эффективность покупки этого региона рассчитывается как $7/9$

Второй регион поля для покупки

Левый верхний угол с координатами [3, 1]

Правый нижний угол с координатами [4, 3]

Его площадь 6, а плодородных участков на нем 3.

Эффективность покупки этого региона рассчитывается как $3/6$

$7/9 > 3/6$, поэтому Василию стоит купить первый регион.

Пример 2

Ввод:

```
5 3
1 1 1 0 1
1 1 1 0 1
1 1 1 0 1
```

Вывод:

9

Здесь эффективность регионов одинакова – они оба полностью заполнены плодородной землей, но регион слева больше, поэтому ответ 9

Примечания по оформлению решения

Возможно использование только стандартных библиотек языков, установки и использование дополнительных библиотек невозможны.

Проверка входных данных не требуется, все данные гарантированно соблюдают условия, указанные в разделе входные данные

При отправке решений на Java необходимо назвать исполняемый класс `Main`. В решении не нужно указывать пакет.

Примеры работы со стандартными потоками ввода и вывода

Для JS можно использовать `readline` и `console.log`:

```
const readline = require('readline').createInterface(process.stdin, process.stdout);
readline.on('line', (line) => {
    // Введенная строка в переменной line, тут можно написать решение и вывести его с помощью console.log
    ...
    console.log(String(result));
    readline.close();
}).on('close', () => process.exit(0));
```

в Python можно использовать встроенные функции `input()` и `print()`:

```
line = input()
...
print(result)
```

в Java можно использовать `java.util.Scanner` и `System.out.println`:

```
Scanner in = new Scanner(System.in);
String line = in.nextLine();
...
System.out.println(result);
```

Перед отправкой решения рекомендуем запустить тесты из раздела *Тестирование*, они помогут поймать синтаксические ошибки и ошибки выполнения.