Кръг V. Думи

# Мечо Пух много обича да играе футбол. И да *tweet*-ва. И да гледа клипчета в *youtube*. Днешната му страст обаче не е нито едно от тези неща, а игрите с думи. Приказвайки с дядо си (старият Бай Пух), Пух разбра за древна, но интересна игра с думи. В миналото хората (и мечките очевидно ☺) са я играели, чертаейки правоъгълно поле в пясъка. След това всеки от играчите изигравал всички свои ходове индивидуално и накрая се правело класиране. Играта протичала по следния начин:

1. На играча се дава информация за размерите на полето (*N* x *M*) и броя на ходовете (*Q*), които играча ще може да направи.
2. На всеки ход от играта, играчът можел да направи няколко избора:
   1. Да поиска следващата дума и да я постави някъде по полето, така че да няма никакви конфликти с думи, поставени до този момент. Тъй като на полето понякога имало камъни, тези клетки се считали блокирани и на тях не можело да се слагат никога букви. Всеки втори път, когато не можели да сложат думата никъде по полето, играчите губели *3* точки.
   2. Да пропусне следващата дума, но вместо да я поставя, да се откаже от нея и да види следващите 5 думи. Легендарните играчи като Бай Пух знаели винаги кога да се възползват от тази възможност и се предполага, че това бил ключа към тяхното майсторство…
   3. Тъй като понякога било много удобно просто да се разкара някой камък, създателите на играта решили реално играчите да имат такава възможност – правейки този избор, играч можел да премести камък от дадено поле в друго, свободно такова (т.е. на което няма друг камък или буква).
3. Резултатът на играча се изчислявал като към сумата на дължините на поставените думи се прибавела сумата на **Cij** за всяка двойка (i, j), описваща поле от дъската, на което има буква, и където **Cij** е с 1 по-малко от броя на думите, в които това поле участва.

Всичко това се стори изключително интересно на Пух, и когато неизбежно дядо му започна да приказва за други отегчителни теми (историите от младостта на Бай Пух са познато до болка на племенника му отдавна ☺), Мечо започна да мисли как би могъл да използва новите технологии, за да стане дори по-добър играч от дядо си. Бързо обаче, той видя, че задачата съвсем не е тривиална и затова реши да се допита до състезателите в *CodeIT*, защото по негово мнение няма по-готини от тях ☺

# Вход:

В тази задача няма да има входен файл. Вашата програма ще взема информацията за играта (и ще играе ходовете) през *Web Service*. За повече информация за това какви методи ще може да извиквате, за да вземете размерите на полето, броя на думите, ходовете и да играете всеки един от ходовете вижте файла ***interface.docx***.

# Изход:

Както при входа, изходен файл няма да има. Вашата програма ще играе ходовете през *Web Service* интерфейса, описан в ***interface.docx.***

# Ограничения:

2 ≤ **R, C** ≤ 500

1 ≤ **Q** ≤ 50 000

1 ≤ **дължина на дума** ≤ 50

# Оценяване и тестване:

По време на кръга на официалния сайт на състезанието ще бъде качен файл ***grading.docx*, който ще дава допълнителна информация за крайното оценяване на решенията на участниците, както и за финалните тестове.**

**Времето за изпълнение на програмата на един тест е 10 секунди.**

# Примерно разиграване:

|  |  |  |  |
| --- | --- | --- | --- |
| Ход: | **Извиквания:** | **Резултат:** | **Обяснение:** |
|  | **Init(1);** |  | Започва нова игра върху тест с номер 1. Номерът на теста, върху който ще се изпълнява Вашата програма, можете да намерите във файла **crossword.in** |
|  |  |  | Следва предаването на информацията за текущата игра: |
|  | **getRowCount();** | 6 | N = 6 |
|  | **getColumnCount();** | 4 | M = 4 |
|  | **getQueryCount();** | 9 | Q = 9 |
|  | **getField();** | {“0000”,  “0010”,  “0010”,  “0001”,  “0000”,  “0000”} | Първоначалното състояние на игралната таблица, която е с размери N на M. |
|  |  |  | Вече разполагаме с информацията за текущата игра, можем да започнем с първия си ход: |
| 1. | **nextWord();** | “bug” | На първи ход избираме да получим нова дума. |
|  | **placeWord(0, 1, false);** | true | Успешно поставихме новата дума хоризонтално, започвайки от втората клетка на първия ред. |
| 2. | **skip();** | {“ham”,  “batman”,  “east”,  “honey”,  “psy”} | На втори ход избираме да получим допълнителна информация за следващите думи в играта. |
| 3. | **relocate(1, 2, 0, 0);** |  | Решаваме да преместим блокираната клетка на 2-ри ред, 3-та колона в началото на 1-ви ред. |
| 4. | **nextWord();** | “ham” |  |
|  | **placeWord(1, 0, false);** | true | Успешно поставихме новата дума хоризонтално. |
| 5. | **nextWord();** | “batman” |  |
|  | **placeWord(0, 1, true);** | true | Успешно поставихме новата дума **вертикално**. |
| 6. | **nextWord();** | “east” |  |
|  | **placeWord(4, 0, false);** | true | Успешно поставихме новата дума хоризонтално. |
| 7. | **nextWord();** | “honey” |  |
|  | **placeWord(1, 0, true);** | true | Успешно поставихме новата дума вертикално. |
| 8. | **nextWord();** | “psy” |  |
|  | **placeWord(3, 2, true);** | true | Успешно поставихме новата дума вертикално. |
| 9. | **nextWord();** | “pie” | 3.14 |
|  | **placeWord(1, 3, false);** | false | Поставянето на новата дума беше неуспешно, но не губим точки, защото за пръв път не можем да поставим дума. |
|  | **getField();** | {“1bug”,  “ham0”,  “ot10”,  “nmp1”,  “east”,  “yny0”} | Полето в края на играта. |
|  | **EndGame(out time);** |  | Прекратяваме играта и получаваме като резултат спечелените от нас точки. В променливата **time** се записва времето в милисекунди, което е било нужно за комуникация с *Web Service*-а. |