

HyperWords

HyperScience Intern Challenge

Time Limit: 5s. **Memory Limit:** 128MiB

В HyperScience обичаме да играем на бордови игри! Затова решихме да споделим страстта си с вас, като ви предлагаме да поиграем една игра, наподобяваща класическата игра [Scrabble](#). И понеже играта е добре известна, ние играем нейна вариация - Scrabble без дъска (да, точно така: *board game* без *board* ;)). Това е игра за един човек и се състои в следното:

Букви

Играчът разполага с торбичка с **N** плочки с букви и поставка за букви. В началото на играта играчът тегли 7 плочки с букви и ги слага на поставката си. Всяка плочка има асоцииран брой точки, както следва:

- празна плочка (означена с '_'): 0 точки
- 'E', 'A', 'I', 'O', 'N', 'R', 'T', 'L', 'S', 'U': 1 точка
- 'D', 'G': 2 точки
- 'B', 'C', 'M', 'P': 3 точки
- 'F', 'H', 'V', 'W', 'Y': 4 точки
- 'K': 5 точки
- 'J', 'X': 8 точки
- 'Q', 'Z': 10 точки

Ходове

Играчът има право да извършва следните ходове:

- Да напише валидна дума, която не е писал до сега, използвайки плочките от поставката си. Ако сборът на точките от буквите в думата е S , за този ход играчът получава S^D (S на степен **D**) точки, където **D** е параметър. След това играчът тегли толкова букви, колкото е дължината на написаната дума. Ако в торбичката няма останали достатъчно плочки с букви, играчът тегли всички плочки в торбичката.
- Да смени една или повече от буквите си. За да направи този ход, в торбичката трябва да са останали поне толкова букви, колкото играчът сменя. За всяка сменена плочка играчът тегли по една нова от торбичката.
- Да прекрати играта си. В този момент играта приключва. Това е единственият ход, който винаги е валиден и играчът е длъжен да го направи *точно веднъж*.

Речник

Както всяка група, където има хора от повече от едно място, в HyperScience често спорим кои думи са валидни, и кои са измислени (има ли "джанка", няма ли "джанка"). За да няма такива спорове в тази игра, сме подготвили речник с 10,000 английски думи и ще считаме този речник за меродавен. Валидните думи са подмножество на думите в този речник. Самият речник е взет от [тук](#), а валидните думи ще се подават на всеки тест (виж "Вход" по-долу).

Точки

Ако играчът направи *невалиден ход*, ако програмата забие, или използва повече от разрешените 5 секунди и 128 мегабайта памет за тест, играта приключва и играчът получава 0 точки за съответния теста. Невалидни са следните ходове:

- Играчът пише дума, която не присъства във валидните думи;
- Играчът пише дума, за която няма необходимите букви;
- Играчът ползва неправилни букви, за изписването на дадена дума (виж по-долу);
- Играчът пише дадена дума за втори път;
- Играчът сменя повече плочки, отколкото има в торбичката;
- Играчът прави ход след като е прекратил играта;
- Играчът се опитва да смени букви, които не присъстват на поставката му.

Цел

Вашата задача е да напишете алгоритъм, който изкарва колкото може повече точки на тази игра.

Оценяване

За да се избегне различната тежест на различните тестове, броя точки, които ще получите за всеки от тях, се смята пропорционално спрямо най-добрия резултат на някой от участниците за дадения тест. Например, ако вашата програма е изкарала YOURS точки, а най-добрият резултат за теста е BEST, то вие ще получите $(YOURS / BEST)^2$ точки за дадения тест. За да се избегне деление на нула, ако на дадения тест сте изкарали нула точки, то винаги получавате 0 за него. Общият резултат за задачата е сборът на точките ви за всеки тест, сметнати по тази формула.

Предварително и финално класиране

По време на състезанието вашата програма ще се изпълнява върху 20 *предварителни* теста и в реално време ще се обновява *предварително класиране*. Обърнете внимание, че финалното класиране ще се базира на други 100 теста, които са генерирани със същия алгоритъм. По този начин се наказват решения, които *overfit*-ват дадените тестове. Забележете, че е възможно *финалното класиране* да се различава от *предварителното* такава!

Генериране на Тестове

Тестовете за задачата са генерирани по следния начин. Първо се избират стойностите на **D** и **N**, като те са избрани така че:

1. В 10% от тестовите: $10 \leq N \leq 30$, $1.0 \leq D \leq 2.0$
2. В 15% от тестовите: $30 < N \leq 200$, $D = 1.0$
3. В 25% от тестовите: $200 < N \leq 500$, $1.0 \leq D \leq 2.0$
4. В 25% от тестовите: $500 < N \leq 1000$, $D = 1.0$
5. В 25% от тестовите: $1000 < N \leq 10000$, $1.0 \leq D \leq 2.0$

След това се генерират **N** букви на случаен принцип, задаващи буквите в торбичката. Буквите са генерирани със същата вероятност, както в scrabble. Това означава, че вероятността дадена буква да се падне, е:

- 12% за 'E'
- 9% за 'A' и 'I'
- 8% за 'O'
- 6% за 'N', 'R' и 'T'
- 4% за 'D', 'L', 'S' и 'U'
- 3% 'G'
- 2% за 'B', 'C', 'F', 'H', 'M', 'P', 'V', 'W', 'Y', а както и за празната плочка ('_')
- 1% за 'J', 'K', 'Q', 'X' и 'Z'

Генератор и чекер

За ваше удобство е даден примерен [генератор на тестове](#). Предварителните и финалните тестове ще са генерирани със същия алгоритъм. Освен това ви се предоставя [checker](#), който изчислява колко точки изкарвате за даден тест с вашето решение.

Вход

Входните данни за всеки тест се четат от стандартния вход. На първия ред ще стоят разделени с точно един интервал числата **N** и **D** - съответно броят букви в торбичката и степента, на която вдигаме точките за всяка дума. Първото е цяло положително число, а второто е число с плаваща запетая.

На следващия ред има точно **N** символа, задаващи буквите в торбичката, в реда, в който ще ги изтеглите. Всеки символ е главна латинска буква или символа '_', обозначаващ празна плочка.

На следващия ред се задава числото **K** ($1 \leq K \leq 10000$) обозначаващо броят валидни думи. На следващите **K** реда има точно по една дума написана с главни латински букви. Думите са подредени по азбучен ред и сред тях няма повтарящи се думи. Гарантирано е че думите са подмножество на речника споменат по-горе.

Изход

След като прочетете входа за даден тест, вашият алгоритъм трябва да отпечата валидна последователност от ходове. Всеки ход трябва да е отпечатан на отделен ред и ходовете се задават по следния начин:

- "P <word> <letters>": поставя думата <word>, използвайки буквите зададени в <letters>. Тук буквата 'P' задава действието (place word), и между нея и думата, както и между думата и буквите трябва да има точно един интервал. Обърнете внимание, че поради наличието на празна плочка е възможно да можете да напишете по повече от един начин дадената дума и затова се налага да кажете изрично кои букви ползвате. Буквите в <letters> трябва да са изредени в реда, в който се ползват, за да изпишат думата.
- "C <letters>": сменя буквите, изброени в <letters>. Тук отново буквата 'C' задава действието (change letters) и между нея и буквите, които се сменят, трябва да има точно един интервал. Буквите, които се сменят, трябва да са изредени без разделител и трябва да са изредени точно толкова пъти, колкото сменяме съответната буква (да речем, за да сменим три букви 'D', трябва да изредим три D-та). Редът на буквите няма значение.
- "T": прекратява играта (съкращава "terminate"). След тази команда не трябва да има повече команди. В противен случай изходът ще се смята за невалиден.

Последната ви команда задължително трябва да е "T" - в противен случай няма да спечелите точки за съответния тест.

Примерен Вход	Примерен Изход
16 1 ABCDEFGF_ZTHINBDY 3 BAD HIGH XYZ	C CEF P BAD B_D P HIGH HIGH T

В тази игра играчът печели 16 точки. Забележете, че играчът може да изпише думата "BAD" и като използва буквите B, A и D в този ред. По този би спечелил 17 точки. Освен това обърнете внимание, че след като изпише думата "HIGH", играчът има отново букви за думата "BAD", но няма право да я напише за втори път.