2 Homework Assignment – Hangman

We are going to write a program to play the game of Hangman (Dutch: *Galgje*). The player has to find a secret word by guessing single letters. When the secret word contains the guessed letter, it is shown in the word. When it doesn't contain the letter, a point is subtracted. When all the letters of the word haven been guessed before the number of points (usually 10 at the start of the game) is exhausted, the player has won the game. Otherwise the player has lost.

Input

First an arbitrary whole number to start the random number generator (see section *Requirements*). Then the guessed letters separated by whitespace.

Output

- First the line "Type an arbitrary number"
- Before each guess: The secret word in which each unguessed letter is printed as an underscore (`_`). When a letter has been guessed it is shown in each position where it occurs in the word.
- After 10 wrong (where wrong means that the letter does not occur in the secret word) guesses, the line "Unlucky, you lost!" followed by a line with "The secret word was: " and the secret word. Or...
- or when all the letters of the secret word have been guessed, first print the secret word on a line, followed by the line "Well done, you won!".

Extra

With the program as described above you can get a maximum grade of 9. If you want to earn an additional point, check whether the guessed letter has been guessed before. In that case, count the guess as a wrong guess even if it occurs in the secret word, and subtract a point accordingly.

Requirements

Use the provided template. Do not change the name of the class or other names used there. You may temporarily change the set of words it chooses from (`bagOfWords` in the template), but you have to use the provided sequence in the program you submit.

You are **required** to use Java's random number generator `Random`. Follow the template and use `Random randomGenerator = new Random(seed);`, where `seed` is a number (of type `long`) read from input. A random number generator produces a sequence of numbers that are seemingly randomly chosen. `randomGenerator.nextInt(n)` gives a random number between 0 (included) en `n` (excluded). Although seemingly random, the sequence is completely determined by the starting value, the *seed*. This way, we can predict the outcome and properly test your program.

Example 1

```
Type an arbitrary number
3
_______
e
_______
a
_a_____
i
_a__i__
```

```
n
_a__in_
g
_a__ing
b
_a__ing
w
wa__ing
l
wal_ing
k
walking
Well done, you won!
```

Example 2

```
Type an arbitrary number
14
___
e
___
t
___
a
a__
o
a__
i
a__
p
a__
p
a__
r
a__
s
a__
t
a__
u
Unlucky, you lost!
The secret word was: and
```