

# Assignment 2: quiz night

Helen Fagan; Victor Cionca

Implement a quiz game engine.

The game engine loads questions and their answers from a file and asks the questions to the user who needs to provide the answer. The game can provide clues to the user, with the amount of information provided depending on the level of difficulty.

## Details

1. The questions and answers are in a file that is provided as a command line argument to the application.
2. In the QnA file each line has a question and its answer. The answers are always one word. The question always ends in '?' then there is a space and following that the answer. For example:

**What is the capital of France? Paris**

3. The application takes a second command line argument, a number <10 that represents the difficulty level.

The program should

- at start
  - load the questions and answers from the file
  - print out the number of questions in the quiz
- start asking questions
  - for each question, display a clue (if available, see below); the user has one attempt only, and the program is case insensitive (accepts both lower and upper case answers)
  - after each question display the number of questions asked and the number of correctly answered questions so far. Example:  
**Score: 5/7**
- at the end store the results in a file **quiz\_history.txt** by appending to the file one line containing:
  - name of the quiz file
  - number of questions answered correctly
  - total number of questions
  - difficulty level

## Questions and clues

The following difficulty levels should be supported:

1. No clue is given and only a '?' is shown. Example:

**What is the capital of France?**  
**?**

2. A set of blank dashes separated by spaces is displayed, one for every letter in the answer. Example:

What is the capital of France?  
- - - - -

3. The first and the last letters are shown. Example:

What is the capital of France?  
P - - - S

4. Two random letters from the answer are shown in their correct position. Example:

What is the capital of France?  
P - R - -

## Observations

- you do not know in advance how many questions are in the file; it is recommended to use a dynamic data structure to store questions as they are read

## Marking

### Basic marking (up to 70%)

- processing command line parameters (5)
  - file name
  - difficulty level
- reading questions from file
  - opening the file and reading from the file (5)
  - separating questions from answers (5)
  - storing a question and its answer (5)
  - storing all the questions: static (5) or dynamic (10)
- print out number of questions in the quiz (3)
- display a question (3)
- take in the user's word and compare it with the question answer (3)
  - case insensitive (2)
- display a clue based on the difficulty level:
  - level 1, 2, 3, 4 (5 each)
- write the results to file (6)
- keep track of the number of correctly answered questions (3)

### Advanced marking (over 70%)

The following tasks will lead to marks over 70%.

1. The questions are asked in random order. (10)
2. Difficulty level 5: all letters are shown but in random order. (10) Example:  

What is the capital of France?  
R S I P A
3. Difficulty level 6. The type of clue is randomly selected from the 5 types listed above. (5)
4. Store the status (correct/incorrect) of all the questions. When the quiz is complete the user is asked if they want to view the correct answers to the questions that were wrong. (5)