

INST0072: Assessment 2, 2020/21

This assignment counts as 25% of the total course assessment.

[[module home page](#)]

Important: Read This First

This assessment forms part of your degree assessment. It must be done **entirely on your own** from start to finish:

- **You must not** collaborate or work with other students at any stage.
- **You must not** send or show other students your answers.
- **You must not** ask other students for help, or ask to see their answers. This is unfair to the other student concerned, since it may lead to them being accused of plagiarism.
- **You must not** seek help from friends, relatives, online discussion groups, or anyone other than the lecturer and/or lab demonstrators for INST0072.
- **You must not** include any automatically generated Prolog code in your answer.
- **You must not** include any Prolog code taken from code libraries in your answer.

If you are having difficulty in attempting this assessment, you should discuss this in the first instance with one of the INST0072 lecturers.

The temptation to plagiarise can often be avoided by **starting the assessment in good time**, allowing plenty of time for testing, debugging and re-testing of your programs, and setting a personal deadline that is well before the submission deadline.

Submission Instructions and Deadline

Please submit this assessment electronically, following the instructions on Moodle at <https://moodle.ucl.ac.uk/mod/assign/view.php?id=2190063> before **3pm, Wednesday 16 December 2020**. Please check that each of the .pl files you are submitting starts with the following in a program comment: (i) your student number (SRN - you will find this on your ID card), (ii) the module code and title, (iii) the lecturer's name: "Dr Rob Miller", and (iv) an indication of which assignment or part of assignment your file concerns. The name of each file that you submit must begin with your student number (SRN - this can be found on your ID card) and the module code, in that order, without spaces - e.g. "123456_INST0072cw2partA.pl".

Mark Distribution, Marking Criteria and Marking Policy

This assignment counts as **25%** of the total course assessment. Part A is worth 12% and B is worth 13%. Marks will be awarded according to:

- whether the programs work correctly on a range of ground, partially ground and/or unground queries, including but not restricted to those queries explicitly given in the questions,
- whether the code is as straightforward as possible and not overly complicated,

- whether the programs are written in a good style, following the guidelines and techniques covered in the lectures and the associated reading,
- whether the programs are set out in a clear and consistent way, as exemplified by the programs in the lecture notes and course texts, and
- whether the programs are sensibly (and not excessively) commented.

Each submission will be first marked according to the criteria given above, and a sample of submissions will also be second marked, using open and check marking, in accordance with the guidelines in [Chapter 4, Section 7.6.1.4\(a-ii, b-ii, c-ii\) of the UCL Academic Manual](#).

Deadline for Feedback and Provisional Mark

This assignment will be marked and returned to you with feedback and a provisional mark on or before **Monday 18 January 2021**.

Part A (12%)

<https://powcoder.com>

In a simplified game of Scrabble, the score for a particular word is always calculated by simply adding up the points for each letter in the word. The points for each letter in the alphabet are represented in the following Prolog program as 26 facts:

Assignment Project Exam Help
Assignment Project Exam Help
Add WeChat powcoder
<https://powcoder.com>
Add WeChat powcoder

```
scrabble_points(a, 1).
scrabble_points(b, 3).
scrabble_points(c, 3).
scrabble_points(d, 2).
scrabble_points(e, 1).
scrabble_points(f, 4).
scrabble_points(g, 2).
scrabble_points(h, 4).
scrabble_points(i, 1).
scrabble_points(j, 8).
scrabble_points(k, 5).
scrabble_points(l, 1).
scrabble_points(m, 3).
scrabble_points(n, 1).
scrabble_points(o, 1).
scrabble_points(p, 3).
scrabble_points(q, 10).
scrabble_points(r, 1).
scrabble_points(s, 1).
scrabble_points(t, 1).
scrabble_points(u, 1).
scrabble_points(v, 4).
scrabble_points(w, 4).
scrabble_points(x, 8).
scrabble_points(y, 4).
scrabble_points(z, 10).
```

Add a definition for the predicate 'scrabble_score/2' to this program, that has a list of lowercase letters (representing a word) as the first argument and a number as the second argument. The query '?- scrabble_score(*Word*, *Score*)' should succeed if and only if *Score* is the Scrabble score for the word represented in the list *Word*. Your answer should reproduce the following example input/output:

```
?- scrabble_score([s, c, r, a, b, b, l, e], Scrabble_score).
Scrabble_score = 14.
```

Part B (13%)

Write a program that includes a definition for the predicate 'word_with_replacements/2'. This predicate should be true if the two arguments are lists, and the second list is the same as the first but with all elements that are the single letter 'a' replaced by the letter 'e', and with all elements that are the single letter 'e' replaced by the letter 'a'. Your answer should reproduce the following example input/output:

```
?- word_with_replacements([s, c, r, a, b, b, l, e], Word_with_replacements).  
Word_with_replacements = [s, c, r, e, b, b, l, a] ;  
false.
```

```
?- word_with_replacements(Word, [s, c, r, e, b, b, l, a]).  
Word = [s, c, r, a, b, b, l, e] ;  
false.
```

(Ideally, the program should be written so that it would be very easy at a later date to specify that more letters should be replaced with others, simply by adding facts to the program.)

<https://powcoder.com>

Page maintained by [Rob Miller](#). Last updated: 28 August 2020.

Assignment Project Exam Help
Assignment Project Exam Help
Add WeChat powcoder
<https://powcoder.com>

Add WeChat powcoder