CISC 360 Assignment 4
due Friday, 2022–11–25 at 11:59pm, via onQ

Jana Dunfield

November 17, 2022

Reminder: All work submitted must be your own, or, if you are working with one other student, your teammate's.

Late policy: Assignments submitted up to 24 hours late (that is, by 11:59 pm the following day) will be accepted **without penalty**. Assignments submitted more than 24 hours late will **not** be accepted, except with an accommodation or a consideration.

If you choose to work in a group of 2

Assignment Project Exam Help

You **must** use version control (such as GitHub, GitLab, Bitbucket, etc.). This is primarily to help you maintain an equitable distribution of work, because commit logs provide information about the members' level of contribution.

https://powcoder.com

Your repository **must** be private—otherwise, anyone who has your GitHub (etc.) username can copy your code, which would violate academic integrity. However, upon request from the course staff, you must give us access to your repository. (But you do not need to give us access unless we ask.)

We only need *one* submission of the assignment ("Assignment 4"). However, each of you *must* submit a brief statement ("Assignment 4 Group Statements").

Add WeChat powcoder

1. Estimate the number of hours you spent on the assignment.

2. Briefly describe your contribution, and your teammate's contribution. (Coding, trying to understand the assignment, testing, etc.)

This is meant to ensure that both group members reflect on their relative contributions.

If you do not submit a statement, you will not receive an assignment mark. This is meant to ensure that each group member is at least involved enough to submit a statement. **Each** member must submit a statement. That is, you must make two separate submissions in "Assignment 4 Group Statements".

IMPORTANT: Your file must compile

Your file **must** load (`consult` in SWI-Prolog) without errors, or we will subtract **30%** from your mark.

If you are halfway through a problem and run out of time, **comment out the problematic code** by surrounding it with `/*...*/` and add a comment describing what you were trying to do. We generally give (partial) marks for evidence of progress in solving a problem, but **we need the file to load without errors**.

It is your responsibility to submit the right version of the file.

(Warnings about singleton variables do not count as errors.)

1 Add your student ID

Begin by adding your student ID number in `a4.pl`, after `student_id(`, replacing "`this is a syntax error`".

```
/*
 * Q1: Student ID
 */
student_id( this is a syntax error ).
% second_student_id(  ).
% If in a group, uncomment the second_student_id line
%   and put the second student's ID between the ( )
```

Assignment Project Exam Help

2 Q2: Prime numbers

https://powcoder.com

The file `a4.pl` contains some of the code necessary to compute prime numbers:

- a predicate `factors(N, Factors)` which returns (in `Factors`) a list of the numbers between 2 and `N - 1` that evenly divide `N`; for example, `factors(20, [2, 4, 5, 10])` is true.

To finish the job, you need to define three predicates:

Add WeChat powcoder

- **Q2a**: `isPrime`
- **Q2b**: `findPrimes`
- **Q2c**: `primes_range`

2.1 Q2a: `isPrime`

Define a predicate `isPrime`, which "returns" 'prime' if a natural number is prime (for example, `isPrime(7, prime)`), and 'composite(*PrimeFactors*)' if composite.

For example, `isPrime(20, composite([2, 5]))` should be true.

2.2 Q2b: `findPrimes`

Given a list of integers `Numbers`, the predicate `findPrimes` should "return" a list of the integers in `Numbers` that are prime.

2.3 Q2c: `primes_range`

Use `upto` (defined in `a4.pl`) and `findPrimes` to define a predicate `primes_range` that finds all the prime numbers in a certain range.

3 Q3: spiral

Translate the function `spiral` from Assignment 1. The file `a4.pl` includes a sample Haskell solution.

4 Q4: Trees

Follow the instructions in `a4.pl`.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder