

CANDIDATE: please
attach Student
Support Unit sticker,
if relevant

G6021

THE UNIVERSITY OF SUSSEX

INFORMATICS

BSc FINAL YEAR EXAMINATION 2021
MComp THIRD YEAR EXAMINATION 2021

January 2021 (A1)

Comparative Programming

Assignment Project Exam Help
Candidates should answer TWO questions out of THREE. If all three
questions are attempted only the first two answers will be marked.

<https://powcoder.com>
Each question is worth 50 marks.

Add WeChat powcoder
Write your answers on A4 paper, scan and save as a single PDF file and
upload to Canvas

PDF file name: candidate number_module title
Read Academic Integrity Statement

You are reminded that, unless you have been authorised to do so in School or
specific assessment guidance, you should not access online materials, notes etc.
during this examination or discuss this assessment with others before the end of
its 24 hour window. By submitting this assessment you confirm that you have read
the above Statement and are responsible for understanding and complying with
our academic misconduct regulations (found on Student Hub and here: Academic
Misconduct regulations).

1. Consider the following program written in Haskell syntax:

```
sq x = x*x
twice (f,x) = f(f(x))
inf x = inf (x+1)
```

- (a) Draw the reduction graph for `twice (\x->x,3+4)`. Underline all redexes. [15 marks]
- (b) Describe in one sentence what is meant by the most general type of a function. For each of the functions: `sq`, `twice` and `inf`, give the most general type. [15 marks]
- (c) Are the following statements true? Give a one-sentence justification for each.
- i. `inf (inf 0)` is a well-typed expression.
 - ii. `inf (inf 0)` will terminate with call-by-name strategy.
 - iii. `sq 4 + inf 4` is a well-typed expression.
 - iv. All well-typed Haskell programs terminate. [10 marks]
- (d) Write a PCF function to add two numbers. Include all types in your answer. [10 marks]

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

2. (a) i. For the following two types, draw the type trees and find the most general unifier, if one exists.

$$\begin{aligned} (A \rightarrow B) \rightarrow (B \rightarrow C) \rightarrow A \rightarrow C \\ D \rightarrow E \rightarrow D \end{aligned}$$

[10 marks]

- ii. Define a function in Haskell that has the most general type:

$$(A \rightarrow B) \rightarrow (B \rightarrow C) \rightarrow A \rightarrow C$$

[10 marks]

- iii. Give the un-curried version of the function in Question 2(a)ii. Include the Haskell code and the type of this function. [10 marks]

- (b) Explain why Prolog would fail to find a solution to the following program, and suggest two ways in which this can be resolved.

Assignment Project Exam Help

```
even(s(s(X))) ← even(X),
even(0),
?even(Y).
```

<https://powcoder.com>

[10 marks]

- (c) What is the *occurs check*? Give an example of the occurs check in both type reconstruction and Prolog evaluation. [10 marks]

Add WeChat powcoder

3. (a) Consider the following Haskell data type:

```
data Tree = Empty | Node Int Tree Tree
```

Write Haskell functions for the following operations:

- i. `mapTree`: apply a function to each element of the tree. Give the most general type of your function. [10 marks]
 - ii. `flatTree`: Convert a `Tree` into a list. You can use `append` (`++`) in your definition. Give the most general type of this function. [10 marks]
 - iii. `flatTreeAcc`: Convert a `Tree` into a list using an accumulating parameter. You cannot use `append` (`++`) in your definition. Give the starting value for the accumulating parameter, and give the most general type of this function. [10 marks]
- (b) Write Prolog clauses to convert a `Tree` data type into a list. You may use the `append` program in your answer. [10 marks]
- (c) Write Java classes to represent the `Tree` data structure. Compare Java, Prolog, and Haskell for 1) representing this data type, 2) writing operations over this data type. [10 marks]

<https://powcoder.com>

Add WeChat powcoder