COMPSCI 1JC3 Midterm Test 1
McMaster University

Answer Key: Large arrow ($\Longleftarrow$) for correct, small ($\leftarrow$) for partially correct

Day Class 01, 02, **Version 1** Dr. W. M. Farmer

DURATION: 2 hours October 20, 2017

Please CLEARLY print:

NAME:

Student ID: | | | | | | | | | |

In an addition to this examination paper, you will be given two answer sheets for this test. This examination paper includes 16 pages and 30 questions. You are responsible for ensuring that your copy of the examination paper is complete. Bring any discrepancy to the attention of your invigilator.

The examination will be conducted in two stages:

First Stage: You have 90 minutes to answer the questions in the examination paper on the first answer sheet working **by yourself**. Getting any help in any form from your fellow students and anyone else will be treated as academic dishonesty. You must submit your first answer sheet to your invigilator by the end of the 90-minute period. Your performance on the answer sheet counts for 85% of the Midterm Test 1 mark. You may want to fill out the second answer sheet as you fill out the first leaving blank those questions that you want to work on during the second stage.

Second Stage: You have 30 minutes to answer the questions in the examination paper on the second answer sheet working **with the other students in the test room**. You may walk around the test room, but you may not leave the test room. You must submit your second answer sheet and your examination paper to your invigilator by the end of the 30-minute period. Your performance on the answer sheet counts for 15% of the Midterm Test 1 mark.

Special Instructions:

(1) It is your responsibility to ensure that the two answer sheets are properly completed. Your examination result depends upon proper attention to these instructions:

- A heavy mark must be made, completely filling the circular bubble, with an HB pencil.
- Print your name, student number, course name, course number and the date in the space provided on the top of Side 1 and fill in the corresponding bubbles underneath.
- **Fill in the bubble corresponding to your version number.**
- Mark only **ONE** choice from the alternatives (1, 2, 3, 4, 5 or A, B, C, D, E) provided for each question. If there is a True/False question, mark 1 (or A) for True, and 2 (or B) for False. The question number is to the left of the bubbles. Make sure that the number of the question on the scan sheet is the same as the number on the examination paper.

- Pay particular attention to the "Marking Directions" given on the scan sheet.
- Begin answering the questions using the first set of bubbles, marked "1." Answer all questions.

(2) The use of notes and textbooks is **not** permitted in both stages of the test.

(3) Calculators, computers, cell phones, and all other electronic devices are **not** to be utilized in both stages of the test.

(4) Read each question carefully.

(5) Try to allocate your time sensibly and divide it appropriately between the questions.

(6) Select the **best** answer for each question.

Question 1 [1 mark]
The functions `fun1` and `fun2` defined below have the same type.

```
fun1 x y = x + y

fun2 (x,y) = fun1 x y
```

Is this statement true or false?

A. True.

B. False. $\Longleftarrow$

ANSWER:
`fun1` has type `Num a => a -> a -> a`, while `fun2` has type `Num a => (a, a) -> a`.

Question 2 [**1 mark**]
In the context of the Haskell code

```
bf1 :: Bool -> Bool -> Bool
bf1 False _ = True
bf1 _ True  = False

bf2 :: Bool -> Bool -> Bool
bf2 _ True  = False
bf2 False _ = True
```

the expression

```
bf1 a b == bf2 a b
```

evaluates to `True` whenever `a` and `b` are boolean values.
Is this statement true or false?

A. True.

B. False. ⟸

ANSWER:
`bf1 False True` evaluates to `True`, while `bf2 False True` evaluates to `False`.

Question 3 [**1 mark**]
Unicode is a scheme for representing the graphemes of the American typewriter. Is this statement true or false?

A. True.

B. False. ⟸

ANSWER:
Unicode is a scheme for representing the graphemes of any writing system. ASCII is a scheme for representing the graphemes of the American typewriter.

Question 4 [**1 mark**]
In Haskell, all type errors in an expression are found before the expression is evaluated. Is this statement true or false?

A. True. ⟸

B. False.

ANSWER:
A Haskell expression can only be evaluated if it is free of type errors.

Question 5 [1 mark]
It is impossible for a computer running the Linux operating system to execute a program written for the Windows operating system. Is this statement true or false?

A. True.

B. False. $\Longleftarrow$

ANSWER:
A program written for Windows can be executed on a computer running Linux by using a Linux program simulating Windows to execute it.

Question 6 [1 mark]
It is impossible to define the boolean function `not` using just the the boolean functions `and` and `or`. Is this statement true or false?

A. True. $\Longleftarrow$

B. False.

ANSWER:
$\{\texttt{and}, \texttt{or}\}$ is a not a complete set of boolean functions in the sense that all other boolean functions can be defined using members of the set.

Question 7 [1 mark]
_____ and _____ showed that _____'s dream of a computer that can decide the truth or falsity of any scientific statement is an impossibility. Which answer fills in the blanks in order correctly?

A. Alonzo Church, Gottfried Leibniz, Alan Turing.

B. Alonzo Church, Alan Turing, Gottfried Leibniz. $\Longleftarrow$

C. Gottfried Leibniz, Alonzo Church, Alan Turing.

D. Alan Turing, Gottfried Leibniz, Alonzo Church.

ANSWER:
Only answer C makes historical sense.

Question 8 [1 mark]
Ada Lovelace, often considered the first computer programmer, wrote programs for a mechanical computer devised by _____.

A. Charles Babbage. ⟸

B. George Boole.

C. Lord Byron.

D. Alan Turing.

ANSWER:
Ada Lovelace wrote algorithms for Charles Babbage's Analytical Engine.

Question 9 [1 mark]
The 8-bit string `01110101` represents the positive integer 117 in the two's complement scheme. What 8-bit string represents −117?

A. `11110101`.

B. `10001010`.

C. `10001011`. ⟸

D. None of the above.

ANSWER:
The answer is obtained by inverting the bits of `01110101` and then adding 1.

Question 10 [1 mark]
The *fundamental question of computational thinking* is concerned with

A. The theoretical and practical limits of computing. ⟸

B. The time and space efficiency of computer programs.

C. The future demand for computing professionals.

D. The best methods for educating computer scientists.

ANSWER:
See slide 11 of the 01 What is Computational Thinking? lecture.

Question 11 [1 mark]
What is strongly avoided in a pure functional programming language like Haskell?

A. Functions with side effects.

B. Code that changes the state of a program.

C. The modification of data.

D. All of the above. $\Longleftarrow$

ANSWER:
A, B, and C are all attributes of imperative programming.

Question 12 [1 mark]
A Turing machine is

A. A mathematical model of computation. $\Longleftarrow$

B. An electronic computer with a tape for processing input and output.

C. A machine with infinitely many internal states.

D. All of the above.

ANSWER:
A Turing machine is a abstract model of computation, not a physical machine.

Question 13 [1 mark]
Modular arithmetic is used by computers to add and multiply

A. Floating point numbers.

B. Machine integers. $\Longleftarrow$

C. Rational numbers.

D. All of the above.

ANSWER:
Modular arithmetic is "clock arithmetic" over a finite initial segment of the natural numbers.

Question 14 [1 mark]
A while loop is an example of a(n)

A. Boolean expression.

B. Conditional expression.

C. Declarative statement.

D. Imperative statement. $\Longleftarrow$

ANSWER:
A while loop is a statement that says to repeatedly execute a body of code as long as a certain condition is true.

Question 15 [1 mark]
What is the hexadecimal number A9C in decimal?

A. 31.

B. 1102.

C. 2716. $\Longleftarrow$

D. 100912.

ANSWER:
$(\text{A9C})_{16} = 10 * 16^2 + 9 * 16^1 + 12 * 16^0 = 2560 + 144 + 12 = 2716.$

Question 16 [1 mark]
What is the decimal number 305 in binary?

A. 10010001.

B. 10011001.

C. 100110001. $\Longleftarrow$

D. 101100010.

ANSWER:
$305 = 256 + 32 + 16 + 1 = 2^8 + 2^5 + 2^4 + 2^0.$

Question 17 [1 mark]
Which number can be represented exactly as a member of the type `Float`?

A. 0.100

B. 0.320.

C. 0.625. $\Longleftarrow$

D. 0.842.

ANSWER:
$0.625 = 0.5 + 0.125 = 1/2 + 1/8 = 1/2^1 + 1/2^3$.

Question 18 [1 mark]
A ripple adder for adding two 16-bit binary numbers would contain _____ half adders.

A. 2.

B. 8.

C. 32. $\Longleftarrow$

D. 34.

ANSWER:
A ripple adder for 16-bit numbers has 16 full adders each composed of 2 half adders.

Question 19 [1 mark]
Which Haskell type contains infinitely many values?

A. `Bool`.

B. `Double`

C. `Float`.

D. `Integer`. $\Longleftarrow$

ANSWER:
`Integer` has a representation for each integer.

Question 20 [**1 mark**]
In the context of the Haskell code

```
x :: Float
x = 32.48
```

what is the type of `[x, x - x]`?

A. `[Float]`. ⟸

B. `[Float, Int]`.

C. `[Float, Integer]`.

D. `[Float, Float]`.

ANSWER:
`[x, x - x]` is a list of values of type `Float`.

Question 21 [**1 mark**]
In Haskell, what is is the type of 53?

A. `Float`

B. `Int`.

C. `Integer`.

D. `Num a => a`. ⟸

ANSWER:
The type of 53 is polymorphic; it can be used as value of any numeric type such as `Integer` or `Float`.

Question 22 [1 mark]
Which function computes the greatest common divisor of two positive integers?

A.
```
paris m n
    | m == n = 0
    | m > n  = paris (m - 1) n
    | m < n  = paris m (n - 1)
```

B.
```
dublin m n
    | m == n = m
    | m > n  = dublin (m - n) n
    | m < n  = dublin m (n - m)
```

⟸

C.
```
london m n
    | m >= n  = london (m - n) n
    | m <= n  = london (n - m) m
```

D.
```
rome m n
    | m == n = m + n
    | m /= n = rome (m - n) (n - m)
```

ANSWER:
`dublin` is Euclid's GCD algorithm.

Question 23 [**1 mark**]
Consider the following function:

```
vilnius m n f
  | m > n   = []
  | m <= n  = vilnius m (n - 1) f ++ f n
```

What is the value of

```
vilnius 1 5 (\x -> [x])
```
?

A. `15`.

B. `[15]`.

C. `[1,2,3,4,5]`. ⟸

D. `[[1],[2],[3],[4],[5]]`.

ANSWER:
`vilnius` computes iterated concatenation of lists like $\sum_{i=m}^{n} f(i)$ computes iterated addition of numbers.

Question 24 [**1 mark**]
Consider the following function:

```
monaco k m n
  | k == 0 = m
  | k > 0  = monaco (k - 1) n (m + n)
```

What is the value of

```
monaco 7 0 1
```
?

A. 8.

B. 13. ⟸

C. 17.

D. 24.

ANSWER:
`monaco 7 0 1` computes the seventh Fibonnaci number.

Question 25 [**1 mark**]
Consider the following function:

```
madrid x
  | x < 0  = x - 1
  | x == 0 = x
  | x > 0  = x + 1
```

What is the value of

```
map madrid [0,2,-3,6,0]
```
?

A. `[-1,1,-4,5,-1]`.

B. `[0,2,-3,6,0]`.

C. `[0,3,-4,7,0]`. ⟸

D. `[1,3,-2,7,1]`.

ANSWER:
`map madrid [0,2,-3,6,0]` returns the list obtained by applying `madrid` to each value of the list `[0,2,-3,6,0]`.

Question 26 [**1 mark**]
Consider the following function:

```
copenhagen :: Integer -> Integer -> Integer
copenhagen x y =
  let z = x - y
  in z^2 + z + 1
```

What is the value of

```
copenhagen 2 3
```
?

A. -2.

B. 0.

C. 1. ⟸

D. 2.

ANSWER:
Just plug and chug.

Question 27 [1 mark]
Consider the following functions:

```
budapest x = x ^ 2

athens x = x ^ 3

vienna f g = \x -> f x * g x
```

What is the value of

```
(vienna budapest athens) 2
```

?

A. 12.

B. 32. ⟸

C. 54.

D. 60.

ANSWER:
Again just plug and chug.

Question 28 [1 mark]
A *prime number* is a natural number greater than 1 whose only factors are 1 and itself. Consider the Haskell code

```
factors n = [m | m <- E1, n E2 m == 0]

isPrime n = factors n == E3

primes n = [p | p <- [1..n], isPrime p]
```

where E_1, E_2, and E_2 are unspecified expressions. What should these three expression be so that

```
primes 100
```

returns a list of the prime numbers less or equal to 100.

A. `[1..n-1]`, `` `div` ``, `[n]`.

B. `[1..n]`, `` `div` ``, `[1]`.

C. `[1..n]`, `` `mod` ``, `[1,n]`. $\Longleftarrow$

D. `[2..n-1]`, `` `mod` ``, `[1,n]`.

ANSWER:
`factors` computes the factors of a natural number; `isPrime` checks that the factors of a natural number `n` includes only 1 and `n`; and `primes` generates the list of primes less or equal to a natural number.

Question 29 [1 mark]
Consider the following boolean function:

```
lisbon a b c =
  if a == True
  then if b == False then c else False
  else if b || c == False then True else False
```

Which of the following truth tables specifies the semantics of `lisbon`?

A.

a	b	c	lisbon a b c
F	F	F	T
F	F	T	F
F	T	F	T
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	F
T	T	T	T

B.

a	b	c	lisbon a b c
F	F	F	T
F	F	T	F
F	T	F	T
F	T	T	F
T	F	F	T
T	F	T	T
T	T	F	T
T	T	T	F

C.

a	b	c	lisbon a b c
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	F
T	F	T	F
T	T	F	F
T	T	T	F

D.

a	b	c	lisbon a b c	
F	F	F	T	
F	F	T	F	
F	T	F	T	
F	T	T	T	$\Longleftarrow$
T	F	F	F	
T	F	T	T	
T	T	F	F	
T	T	T	F	

ANSWER:
Compute the truth table for `lisbon` and then see that it is same as D.

Question 30 [1 mark]
Consider the boolean function `warsaw` specified by the following truth table:

a	b	c	warsaw a b c
F	F	F	T
F	F	T	F
F	T	F	F
F	T	T	F
T	F	F	T
T	F	T	T
T	T	F	F
T	T	T	F

Which of the following function definitions correctly implements `warsaw`?

A.
```
warsaw a b c =
    ((not a) && (not b) && (not c)) ||
    (a && (not b) && (not c)) ||
    (a && (not b) && c)
```

$\Longleftarrow$

B.
```
warsaw a b c =
    ((not a) || (not b) || (not c)) &&
    (a || (not b) || (not c)) &&
    (a || (not b) || c)
```

C.
```
warsaw a b c =
    (a && b && c) ||
    ((not a) && b && c) ||
    ((not a) && b && (not c))
```

D.
```
warsaw a b c =
    ((not a) && (not b) && c) ||
    ((not a) && b && (not c)) ||
    ((not a) && b && c) ||
    (a && b && (not c)) ||
    (a && b && c)
```

ANSWER:
Compute the truth table of each version of `warsaw` and notice that the truth table of the A version is the same as the truth table that specifies `warsaw`.

Please make sure your **version number** is clearly marked on your scan sheet!

THE END.