KIT103/KMA155 Programming Assignment 2: Logic

Enter your answers to these questions in the accompanying Python script file `kit103_assign2.py`. Unlike programming assignment 1, this time many of your answers will be function definitions, so instead of a dictionary to store your answers the script file contains many function 'stubs'. These stubs have a header specifying their name and parameters but currently return dummy values.

Submit your completed script file to the *Programming Assignment 2 (Logic)* dropbox on MyLO by **1500 (3pm) Wednesday 30 August 2017**.

> Test your solutions thoroughly. Your submission is expected to run without failure (even if it doesn't produce the correct answer for each question). If we have to correct your submission in order for it to run then the maximum total mark you can receive will be 3/5 (1.5/2.5 if in KMA155).

KMA155 students will be assessed primarily on questions marked with an asterisk (*), questions 2a–b, 3a–b and 4, but may attempt all questions if they wish.

> **Tip:** You have been provided with a useful function in one of the practical classes to use while testing your solutions to Questions 2–4.

Question 1: Riding the Sine Wave (1.5 marks)

The Fantastic Factorial Fun Park has a popular roller coaster (called the Sine Wave) with a somewhat complex set of rules for who can go on it. They have tasked you with writing a function that can tell people whether they can ride it or not. The function has three parameters: the person's height in whole cm, age in whole years and a Boolean indicating if they are accompanied by an adult. It can return three possible string messages: 'Sorry, you cannot ride' if the rules exclude a person from riding the Sine Wave; 'Find an adult' if the person (child in this case) can ride as long as they come back with an adult; and 'You can ride' if they are allowed to ride the Sine Wave.

The rules are as follows:

- Anyone less than 110cm or taller than 200cm cannot ride
- Anyone younger than 6 cannot ride
- If a child aged between 6 and 9 (inclusive) wishes to ride, they can do so if they are accompanied by an adult, otherwise they are told that they must 'Find an adult'
- Everyone else is allowed to ride the roller coaster

Your task is implement the rules above by completing the implementation of `q1_sine_wave_check` in the assignment script file. It is currently very pessimistic, telling all patrons that they are not allowed on. Complete the implementation by using any combination of if statements and Boolean expressions. The if statements may be nested as you see fit.

> **Note that:**
>
> - the function must **`return`** the appropriate value, *not* print it; and
> - the string messages must be exactly as above.

Full marks will be given to solutions that use as few tests as possible. A solution that works for all cases, but which repeats a test or implements each rule completely separately, will get 1 mark. A solution that works for only many possible inputs will receive 0.5 marks, while a solution that works rarely or not at all will receive no marks.

Sample Test Data

Question 2: Implementing predicates as functions (1 mark)

There are stub (i.e., incomplete) implementations of each of these predicates as functions in the assignment script file (named `q2_a` through `q2_d`). Replace `None` in each of these stub functions with an implementation of the predicates below as they are written. The functions already have the required parameters listed. All parameters are Boolean.

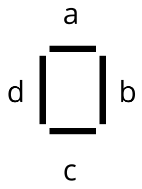
a. (*) $\neg(a \wedge b) \wedge (a \vee b)$
b. (*) $a \vee (\neg b \vee \neg c \vee \neg d)$
c. $(a \vee b) \wedge (a \vee c)$
d. $a \wedge \neg a$

Question 3: Simplifying predicates (1 mark)

For each subpart (a)–(d) in Question 2, write a simplified implementation. There are stub functions named `q3_a` through `q3_d` for you to complete. KMA155 students must attempt at least parts (a) and (b).

Question 4: The Letter Detector (1.5 marks)

The ACME Logic Company has a simple object recognition system that can identify when a particular arrangement of horizontal and vertical lines represents a letter (not which letter, only that it is a letter). They have a camera that can detect four different lines: horizontal lines either at the top or bottom of an image, and vertical lines at the left or right. Although the image may be hand-drawn, their detector ‘sees’ it as if it were like part of a digital display, like this:



To determine if the arrangement of lines looks like a letter they use the following truth table (for convenience the truth table is presented using 0 for False and 1 for True, but the function accepts four Boolean variables a–d and returns a Boolean value). The Letter column is only for information; the function will not attempt to actually identify the letter.

a	b	c	d	Output	Letter
0	0	0	0	0	
0	0	0	1	1	I
0	0	1	0	0	
0	0	1	1	1	L
0	1	0	0	1	I
0	1	0	1	0	

a	b	c	d	Output	Letter
0	1	1	0	0	
0	1	1	1	1	U
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	1	C
1	1	0	0	0	
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	1	O or D

You have two tasks:

a. (*) Transfer the information in the truth table to the Karnaugh map stored in `q4_kmap` in the script file, treating the columns as *ab* and rows as *cd*. `q4_map` is a list of lists, but you can treat it as if it were a table: replace the appropriate locations with 1s. **Tip:** Draw the Karnaugh map on paper first and identify the groups. You won't be able to show the groups in the script file, but this will help with part (b). (0.5 marks)

b. (*) Replace the complicated Boolean expression currently in `q4_acme_letter_detector` with a simplified expression determined from your Karnaugh map. (1 mark)