

KIT103/KMA155 Programming Assignment 1: Sets

Enter your answers to these questions in the accompanying Python script file `kit103_assign1.py`. Each question has an entry in a dictionary of answers that is initially set to `None`. Replace this with the requested Python expression (that is, code that generates a value). For instance, to answer Question 1 part a you could put:

```
ans['Q1 a'] = { 'this', 'is', 'just', 'an', 'example' }
```

Wherever a question indicates that certain values are given and must be used in your answer you will find that these variables are also defined in the script file.

Submit your completed script file to the *Programming Assignment 1 (Sets)* assignment folder on MyLO by **1500 (3pm) Wednesday 16 August 2017**.

Each question part is worth 0.25 marks. KMA155 students will be assessed primarily on questions marked with an asterisk (*), but may attempt all questions if they wish.

Question 1: Literal Values

Write these sets as literal values (i.e., program text that *literally* represents the value described).

- a. (*) the first 6 positive even numbers greater than zero
- b. (*) the names *Bob*, *Alice*, *Donna* and *Charlie* (with that capitalisation)
- c. the set of English vowels (a, e, i, o and u; remember to put them inside quote marks)

Question 2: Set Comprehensions

Write set comprehensions to generate the following sets:

- a. (*) the set { 8, 16, 24, 32, 40 }
- b. (*) cubes of the first 6 positive integers (i.e., { 1^3 , 2^3 , 3^3 , 4^3 , 5^3 , 6^3 })
(tip: Python has an exponentiation operator `**`)
- c. the set of letters in the word 'abracadabra' (hint: strings in Python can be treated exactly like lists of characters). Your solution should still work if we were to change the word, so *do not* remove repeated letters yourself.

Question 3: Relationships

Assume that the following sets have been defined:

```
mammals = { 'cow', 'gorilla', 'human', 'kangaroo' }
aves = { 'cormorant', 'eagle', 'emu' }
quadrupeds = { 'cow', 'lizard' }
bipeds = { 'cormorant', 'eagle', 'emu', 'gorilla', 'human', 'kangaroo' }
can_swim = { 'cormorant', 'cow', 'human', 'kangaroo' }
can_fly = { 'aeroplane', 'cormorant', 'eagle' }
```

Write boolean expressions using Python set comparison operators (i.e., `issubset()`, `<=`, `==`, etc.) and the above sets to answer the questions below. Do not simply say yes or no (imagine that the sets above

were actually complete and so too large for you to answer these questions unaided). Ensure you answer the question as asked.

- a. Are bipeds a proper subset of mammals?
- b. (*) Can all bipeds swim?
- c. Are all aves (birds) also bipeds?

Question 4: Set membership

Given the sets defined in Question 3, identify the following subgroups (as above, write the Python code that will find the answer):

- a. Mammals that cannot swim
- b. Things that can fly and swim
- c. (*) Bipeds that can swim but are not mammals

Question 5: Combinations

Jodie would like to identify her dream vehicle by enumerating various possibilities based on speed, colour and vehicle type. Given the following sets of options, write set comprehensions that will generate the possible combinations described below. Note that the order of items in each combination should match the order in the question.

```
colours = { 'red', 'green', 'blue', 'yellow' }
vehicles = { 'car', 'boat', 'rocket' }
speeds = { 'slow', 'fast' }
```

- a. Generate all pairs of speeds and vehicles.
- b. Generate novel vehicle types by combining vehicles with other vehicles as pairs. These may include the same vehicle twice.
- c. (*) Generate all triples (3-tuples) of speed, colour and vehicle.

Question 6: Bags & Bitsets

- a. Write a dictionary literal that represents the bag/multiset { 'cat', 'hat', 'sat', 'cat', 'sat', 'sat' }

Parts b through e assume the following are defined: a universe of discourse $\mathcal{U} = \{ \text{'alfa', 'bravo', 'charlie', 'delta', 'echo'} \}$, and sets `set1 = { 'delta', 'echo' }` and `set2 = { 'alfa', 'bravo', 'delta' }`.

- b. (*) Write the binary literal value that encodes `set1` as a 5-bit long bitset. (Remember that a binary literal begins with `0b`.)
 - c. (*) Write the binary literal value that encodes `set2` as a 5-bit long bitset.
 - d. (*) Write the binary literal that corresponds to the result of the set operation `set1 \oplus set2`.
 - e. Referring to the bitset representations of `set1` and `set2` stored in your previous answers `ans['Q6 b']` and `ans['Q6 c']`, respectively, write the bit-wise expression that would calculate the result of the set operation `set1 \cup set2`.
-