BIRKBECK

(University of London)

MSc EXAMINATION FOR INTERNAL STUDENTS

MSc Computer Science

MSc Data Science

Department of Computer Science and Information Systems

Principles of Programming I

BUCI033S7

DURATION OF PAPER: One Hour

Practical — Mock Paper — Three

RUBRIC:

1. Candidates should attempt ALL 6 questions on this paper.
2. You are advised to look through the entire examination paper before getting started, in order to plan your strategy.
3. Simplicity and clarity of expression in your answers is important.
4. All programming questions should be answered using the Python programming language.
5. Electronic calculators are **NOT** allowed.
6. Start each question on a new page.

Question:	1	2	3	4	5	6	Total
Marks:	4	8	4	10	10	14	50

Question 1 ..... Total: 4 marks

Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest) and another number. The function decides whether or not the given number is inside the list and returns (then prints) an appropriate boolean.

Question 2 ..... Total: 8 marks

Ask the user for a number and determine whether the number is prime or not. (For those who have forgotten, a prime number is a number that has no divisors.).

Question 3 ..... Total: 4 marks

Write a program that asks the user for a long string containing multiple words. Echo back to the user the same string, except with the words in backwards order. For example, say we type the string:

```
My name is Michele
```

Then we would expect to see the string:

```
Michele is name My
```

Question 4 ..... Total: 10 marks

Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Make sure to ask the user to enter the number of numbers in the sequence to generate. (Hint: The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, ...)

Question 5 ..... Total: 10 marks

Take two lists, say for example these two:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
```

and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes. Write this using at least one list comprehension.

Question 6 ..... Total: 14 marks

Create a program that will play the "cows and bulls" game with the user. The game works like this:

Randomly generate a 4-digit number. Ask the user to guess a 4-digit number. For every digit that the user guessed correctly in the correct place, they have a "cow". For every digit the user guessed correctly in the wrong place is a "bull". Every time the user makes a guess, tell them how many "cows" and "bulls" they have. Once the user guesses the correct number, the game is over. Keep track of the number of guesses the user makes throughout the game and tell the user at the end.

Say the number generated by the computer is 1038. An example interaction could look like this:


```
Welcome to the Cows and Bulls Game!
Enter a number:
>>> 1234
2 cows, 0 bulls
>>> 1256
1 cow, 1 bull
...
```

Until the user guesses the number.

