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

DATE OF EXAMINATION: Tuesday, 29th May 2018
TIME OF EXAMINATION: 2:30pm
DURATION OF PAPER: One Hour

Written — Practice Paper

This paper will be out of 50 marks but we have provided additional practice questions as this is the first time this module has been examined.

RUBRIC:

1. Candidates should attempt ALL XXX 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	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Marks:	24	14	5	3	6	9	12	6	2	20	5	2	2	3	3	9	5	5	12	5	20

Question 1 ..... Total: 24 marks

(a) Write a program that uses loops to print a multiplication table like the one below: 8 marks

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50

Do not concern yourself with aligning columns, but do print a space between numbers, and a newline at the end of each row.

(b) Write a program to list all the *perfect numbers* between 1 and 500. A positive whole number is *perfect* if the sum of its factors (apart from itself) equals the number. 8 marks

For example, the number 6 is perfect because its factors are 1, 2, and 3, and 1+2+3=6.

(c) Write a program for a number guessing game. The program generates a random number (see function below) between 0 and 99, and then asks the user to guess that number. For each guess the program replies 8 marks

- "Correct",
- "Too low", or
- "Too high".

If the number is correct, the program prints the number of guesses it took. If not, the program asks the user to guess again.

For example:

```
Guess a number between 0 and 99: 50
Too low. Guess again: 75
Too high. Guess again: 60
Too high. Guess again: 54
Correct. It took you 4 guesses.
```

Use the following function to choose the random number:

```
int random (int n)
```

that returns a random integer between 0 and `n`.

Question 2 ..... Total: 14 marks

(a) Write a function to search a list of strings for a specified value. The function takes a search string, a list of strings, and the size of the list and returns the number of the slot that contains the search string, or returns -1 if the search string does not appear. 8 marks

```
def search (s, slst, size):
        pass
```

where `s` is the search string, `slst` is the list, and `size` is the number of items in `slst`.

(b) An input file contains exactly 500 student names and their score on anexam, for example, 6 marks

```
JBrown 42 MBlack 55 SPrunty 68 LHughes 36 .
```

Write a program that reads in the data and then repeatedly asks for a name, and returns the score corresponding to that name. The program terminates when the user enters the string `XXX`. For example,

```
Enter a name: LHughes
Grade: 36
Enter a name: MBlack
Grade: 55
Enter a name: XXX
Good bye.
```

You should make use of your `search` function from your previous answer.

Question 3 ..... Total: 5 marks

Write a function that, given a list of integers, returns the first negative number in the list. If there are no negative numbers, it should return `None`.

Question 4 ..... Total: 3 marks

Use a *list comprehension* to set powers to a list containing the first 100 powers of 2 (that is, [1, 2, 4, 8, 16, 32, ...]).

Question 5 ..... Total: 6 marks

Provide an English description of what the following function does:

```
import random
def points(n):
    directory = {}
    for i in range(0, n):
        pname = chr(ord('a') + i)
        x = 1000.0 * random.random()
        y = 1000.0 * random.random()
        directory[pname] = (x, y)
    return directory
```

Question 6 ..... Total: 9 marks

Each of the following pieces of code has an error. State what the error is and how to correct it.

(a) 3 marks

```
# nums is a list of integers. Add the first and last numbers in the list,
# and append the result to the list.
nums = nums.append(nums[0] + nums[-1])
```

(b) 3 marks

```
# Delete negative numbers from the list
for i in lst:
    if lst[i] < 0:
        lst.remove(lst[i])
```

(c) 3 marks

```
# print the average of the three given numbers a, b, c
print "The average is " + (a + b + c) / 3
```

Question 7 ..... Total: 12 marks

(a) Given the following function: 3 marks

```
def foo(s):
    d = {}
    for ch in s:
        v = d.get(ch, 0)
        d[ch] = v + 1
    return d
```

what is returned by the call `foo("sassafras")` ?

(b) Given the following function: 3 marks

```
def letters(s):
    ss = []
    for ch in s:
        if ch.isalpha():
            ss.append(ch.lower())
    return ss
```

what is returned by the call `letters("1 and 2 and 3")` ?

(c) Given the following function: 3 marks

```
def lets(s):
    return "".join(filter(lambda x: x.isalpha(), list(s)))
```

what is returned by the call `lets("1 and 2 and 3")` ?

(d) Given the following function: 3 marks

```
def f(n):
    return [foo, points, letters, lets][n]
```

what is returned by the call `f(1)` ?

Question 8 ..... Total: 6 marks

Suppose you are defining a class `Circle`, and every object of this class must have three values: the `x` and the `y` coordinates of the circle's centre, and the `radius` of the circle.

(a) Write the constructor for this class. 3 marks

(b) Use the above constructor to create a circle named `unitCircle` with radius = 1 and centre at the origin (x = y = 0). 3 marks

Question 9 ..... Total: 2 marks

What is the purpose of the following code in a Python program?

```
if __name__ == '__main__':
    main()
```

Question 10 ..... Total: 20 marks

For each of the following, write the fewest possible Python statements to accomplish the required task. (Half credit if your answer is correct, but longer than necessary.)

(a) Use a for loop to print the contents of list variable `list`, in order, one value per line. [2 marks]

(b) Use a `while` loop to print the contents of list variable `list`, in order, one value per line. [2 marks]

(c) Use a `for` loop to print the contents of list variable `list`, in reverse order, one value per line. [2 marks]

(d) Print one of the words `negative`, `zero`, or `positive`, according to whether variable `x` is less than zero, zero, or greater than zero, respectively. [2 marks]

(e) Create a file named `foo.txt`, and write all the values in list variable `words` to it, one value per line. [2 marks]

(f) Write a function named `isEven` that, given a single integer parameter, returns `True` if the parameter is an even number, `False` otherwise. [2 marks]

(g) Write a unit test function using pytest that says calling `collatz(7)` should return 22. [2 marks]

(h) Write a unit test function using pytest that says calling `evenRand()` should return an even number. [2 marks]

(i) Write a unit test function using pytest that calls `randBuzz()`. The test should pass if `randBuzz()` returns either a positive number or the string `'buzz'`, and fail otherwise. [2 marks]

(j) Create, and save in a variable, a 10x10 array (list of lists), all of whose values are `None`. [2 marks]

Question 11 ..... Total: 5 marks

The following recursive function is supposed to compute the factorial of a positive number (for example, `factorial(5)` = 5*4*3*2*1). It doesn't work. Fix it (do not rewrite the function to use a loop, just fix the error.)

```
def factorial(n):
    return n * factorial(n - 1)
```

Question 12 ..... Total: 2 marks

Inside a function `foo` you see the statement `print(bar)`. How do you figure out whether `bar` is *local* or *global*? Provide a rule, or rules, for determining this.

Question 13 ..... Total: 2 marks

Use a *list comprehension* to assign to the variable `oddCubes` the cubes of the odd integers between 0 and 100.

Question 14 ..... Total: 3 marks

In a method inside a class, what important value is held in the variable `self`? Or, to put the question another way, what is special about this particular value that isn't true about any other variable in the method?

Question 15 ..... Total: 3 marks

There are rules for good programming style, and they may help, but good style isn't about following a bunch of rules. According to what you learnt in module, what is the purpose of those rules? What are they intended to help you achieve?

Question 16 ..... Total: 9 marks

Assume that you have written a class `ListUtils` to provide a number of operations on lists. This class is in the file `listutils.py`. In the following three questions you will write a complete class to test one of the functions in `ListUtils`.

(a) Write the necessary `import` statements and the class header for a class named `ListUtilsTest`. [3 marks]

(b) Write a test for the function `duplicate(lst)` which is supposed to return a shallow copy of the list given as a parameter. [3 marks]

(c) Write the statement or statements necessary to perform the tests in the `ListUtilsTest` class. [3 marks]

Question 17 ..... Total: 5 marks

State whether the following statements about Test Driven Development (TDD) are True or False.

(a) TDD guarantees that your program will be correct.

(b) TDD encourages the use of short, single-purpose functions.

(c) TDD reduces the amount of time spent debugging.

(d) TDD results in more efficient programs.

(e) TDD makes programs easier to modify at some future date.

Question 18 ..... Total: 5 marks

(a) A function named `largest` takes a list of integers as a parameter, and returns the largest number in the list (or `None` if the list is empty). Write a pytest unit test function to test `largest`. [3 marks]

(b) State what you would need to change in the answer to the previous question if `largest` takes a list of floating point numbers instead of integers. (You can rewrite the entire function if that is easiest, or just state what changes need to be made.) [2 marks]

Question 19 ..... Total: 12 marks

For each of the following, write a single (that is, one) Python statement:

(a) Tell the user to `Enter your name`, then read in, and assign to a variable, a string typed in by the user. [1 mark]

(b) Tell the user to `Enter your age`, then read in, and assign to a variable, an integer typed in by the user. [1 mark]

(c) Print out, on separate lines, `Your name is var1` and `Your age is var2`, where `var1` and `var2` are the values of the two variables entered above. [1 mark]

(d) Print out the first character (only) in the string variable `answer`. [1 mark]

(e) Print out the last value (only) in the list variable `classes`. [1 mark]

(f) Create, and save in a variable, a list of all the characters in the string variable `sentence`. [1 mark]

(g) Create, and save in a variable, a new list containing all but the first and last values in the list variable `scores`. [1 mark]

(h) Create, and save in variable `list2`, a copy of the list variable `list1`. [1 mark]

(i) Set the variable `odd` to `True` if the integer variable `count` is an odd number, and `False` otherwise. [1 mark]

(j) Set the variables `sum` and `product` to the sum of `x` and `y` and the product of `x` and `y`, respectively. [1 mark]

(k) Set the variable `mid` to the "middle" value (neither the max nor the min) of the three integer variables `x`, `y`, and `z`. [1 mark]

(l) List variable `list1` is a list of integers. Set variable `list2` to contain all the positive integers in `list1`. [1 mark]

Question 20 ..... Total: 5 marks

State one way in which using TDD, Test Driven Development, may improve the style (not the correctness) of a program. You should provide an example to support your answer.

Question 21 ..... Total: 20 marks

For each of the following functions state what each does. That is, assuming each function is called with the correct type of parameters, state what is computed and returned. Also, if there are side effects, tell what they are. Do not provide a line by line description of how the function does its job, just state what it accomplishes.

(a) [4 marks]

```
def firstBad(aList):
    # aList is a list of integers
    for i in range(0, len(aList)):
        if aList[i] < 0 or aList[i] > 100:
            return i
    return -1
```

(b) [4 marks]

```
def buzz(anInt):
    # anInt is a positive integer
    if anInt % 7 == 0 or '7' in str(anInt):
        return 'buzz'
    else:
        return anInt
```

(c) [4 marks]

```
def flip(aList):
    # aList is a list of numbers
    x = aList[:]
    for i in range(1, len(aList)):
        if x[i] < x[i - 1]:
            x[i], x[i - 1] = x[i - 1], x[i]
    return x[-1]
```

(d) [4 marks]

```
def mapEven(aList):
    # aList is a list of positive integers
    even = map(lambda x: x % 2 == 0
    return list(even, aList))
```

(e) 4 marks

```
def filterEven(aList):
    # aList is a list of positive integers
    f = lambda x: x % 2 == 0
    return list(filter(f, aList))
```