THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2015
Campus: City

COMPUTER SCIENCE

Principles of Programming

(Time Allowed: TWO HOURS)

Note:
- The use of calculators is NOT permitted.

- You should separate the Section A Question Booklet from the Section B Question/Answer Booklet. You may keep the Section A Question Booklet. You must hand in the Section B Question/Answer booklet and the Teleform sheet.

- Compare the exam version number on the Teleform sheet supplied with the version number above. If they do not match, ask the supervisor for a new sheet.

- Enter your name and Student ID on the Teleform sheet. Your name and Student ID should be entered left aligned. If your name is longer than the number of boxes provided, truncate it.

- Answer **Section A** on the Teleform answer sheet provided. For Section A, use a dark pencil to mark your answers in the answer boxes on the Teleform sheet. Check that the question number on the sheet corresponds to the question number in this question/answer book. Do not cross out answers on the Teleform sheet if you change your mind. You must completely erase one answer before you choose another one. If you spoil your sheet, ask the supervisor for a replacement. There is one correct answer per question.

- Answer Section B in the space provided in the Section B Question/Answer Booklet.

- Attempt all questions. Write as clearly as possible. The space provided will generally be sufficient but is not necessarily an indication of the expected length. Extra space is provided at the end of this exam book.

SECTION A

MULTIPLE CHOICE QUESTIONS

For each question, choose the **best** answer according to the information presented in lectures. Select your preferred answer on the Teleform answer sheet provided by shading in the appropriate box.

Question 1
[2.5 marks] Consider the following program:

```
def main():
    sum = 0
    user_num = 1
    while int(input("Enter a number: ")) > -1:
        sum = sum + user_num
    print("Sum:", sum)
main()
```

In an execution of the above program, the user is prompted for a number four times as shown below (the user input is shown in bold and in a larger font). Note that the final line of output is not shown.

```
Enter a number: 5
Enter a number: 0
Enter a number: 2
Enter a number: -1
```

What is the final line of output of the above program execution?

   (a) `Sum: 0`
   (b) `Sum: 7`
   (c) `Sum: 4`
   (d) `Sum: 5`
   (e) `Sum: 3`

Question 2
[2.5 marks] What is the output produced by the following code?

```
sum = 0
for num1 in range(10, 4, -3):
    sum = sum + num1
for num2 in range(1, 5, 2):
    sum = sum + num2
print("Sum:", sum)
```

   (a) `Sum: 25`
   (b) `Sum: 21`
   (c) `Sum: 30`
   (d) `Sum: 10`
   (e) None of the above.

Question 3
[2.5 marks] What is the output of the following program?

```
def fiddle_list(a_list1):
    a_list2 = [2, 3, 4]
    for index in range(0, len(a_list2)):
        a_list1[index] = a_list1[index] + a_list2[index]

def main():
    a_list = [3, 4, 5, 6]
    fiddle_list(a_list)
    print(a_list)
main()
```

   (a) `[5, 7, 9, 10]`
   (b) `[3, 8, 10, 12]`
   (c) `[3, 4, 5, 6]`
   (d) `[5, 7, 9, 6]`
   (e) The program causes the error: `"IndexError: list index out of range."`

Question 4
[2.5 marks] What is the output of the following program?

```
def fiddle_list(a_list1):
    a_list2 = a_list1
    for index in range(0, len(a_list1)):
        a_list1[index] = a_list1[index] + 1
        a_list2[index] = a_list2[index] + 1
    return [1, 2, 3, 4]

def main():
    a_list1 = [3, 4, 5, 6]
    a_list2 = fiddle_list(a_list1)
    print(a_list1)
main()
```

   (a) `[1, 2, 3, 4]`
   (b) `[3, 4, 5, 6]`
   (c) `[5, 6, 7, 8]`
   (d) `[4, 5, 6, 7]`
   (e) None of the above.

Question 5

[2.5 marks] What is the output of the following program?

```
def fiddle_tuples(t1, t2):
    t3 = (t1[1], t2[0])
    t1 = (t3[1], t2[0])
    t2 = t1
    return t3

def main():
    t1 = (3, 5)
    t2 = (4, 7)
    t3 = t2
    t3 = fiddle_tuples(t1, t2)
    print(t1, t2, t3)
main()
```

   (a) (3, 5) (4, 7) (5, 4)
   (b) (4, 4) (4, 4) (5, 4)
   (c) (3, 5) (5, 4) (5, 4)
   (d) (3, 5) (4, 7) (4, 7)
   (e) (4, 4) (4, 4) (4, 7)

Question 6

[2.5 marks] Which of the following Boolean expressions would cause the body of a `while` loop to not execute at all?

   (a) 3 <= 11
   (b) 2 <= 2
   (c) not (7 != 14)
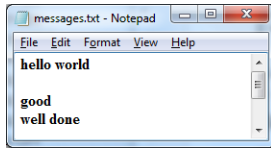   (d) 6 != 9
   (e) None of the above.

Question 7

[2.5 marks] What is the output produced by the following code?

```
num1 = 8
num2 = 4
result = str(num1 + num2) * num2
print(result)
```

   (a) 12121212
   (b) 24
   (c) 12
   (d) 84848484
   (e) None of the above.

Question 8

[2.5 marks] Consider the following text file:

What is the output produced by the following code?

```
infile = open('messages.txt', 'r')
content = infile.read()
words = content.split()
print(len(words), words)
```

   (a) 3 ['hello', 'good', 'well']
   (b) 5 ['hello', 'world', 'good', 'well', 'done']
   (c) 3 ['hello world', 'good', 'well done']
   (d) 4 ['hello world', '', 'good', 'well done']
   (e) None of the above.

Question 9

[2.5 marks] Consider the following function definition:

```
def unknown(first_num, second_num, third_num):
    max_value = max(max(first_num, second_num), third_num)
    return int(max_value) + 1
```

What is the output produced by the following statement?

```
    print(unknown(-23.4, 0.03, 27.3))
```

   (a) 29
   (b) 0
   (c) 1
   (d) 27.6
   (e) 28

Question 10

[2.5 marks] Which of the following statements should be used to draw a rectangle at location (10, 20) with a width 70 and a height 50 on a canvas?

   (a) a_canvas.create_rectangle(10, 20, 60, 90)
   (b) a_canvas.create_rectangle(10, 20, 10, 20)
   (c) a_canvas.create_rectangle(10, 20, 80, 70)
   (d) a_canvas.create_rectangle(10, 20, 70, 50)
   (e) a_canvas.create_rectangle('50x70+10+20')

Question 11
[2.5 marks] Consider the following code fragment:

```
point = (3,7)
animals = ['cat', 'dog']
result = {'John':48, 'Mike':52}
```

Which of the following statements will cause an error to occur?

   (a) `animals.append('mat')`
   (b) `result['Mary'] = 90`
   (c) `animals[1] = 'bat'`
   (d) `result['John'] = 98`
   (e) `point[1] = 4`

Question 12
[2.5 marks] Consider the following dictionary:

```
d = {4: ['mark', 'paul'], 5: ['henry'], 6: ['joseph']}
```

Which of the following statements should be used to print the keys of the above dictionary in a list format, e.g. `[4, 5, 6]`?

   (a) `print(d)`
   (b) `print(list(d.keys()))`
   (c) `print(d.values())`
   (d) `print(d.keys())`
   (e) `for k,v in d.items():`
         `print(k, end="")`

Question 13
[2.5 marks] What is the output produced by the following code?

```
total = 40
num1 = 15
if num1 > 14:
    total = total + num1
elif num1 > 10:
    total = total - num1
else:
    total = total - (2 * num1)
print(total)
```

   (a) `55`
   (b) `40`
   (c) `25`
   (d) `30`
   (e) None of the above.

Question 14

[2.5 marks] Consider the following function definition:

```
def search(river, target):
    ???
        if river[index] == target:
            return 'FOUND'
    return 'NOT FOUND'
```

Which of the following statements should replace the ??? above to correctly complete this function definition which searches for the `target` character in the `river` string variable?

    (a) `for index in river:`
    (b) `for index in range(river):`
    (c) `for index in range(len(river)):`
    (d) `for letter in river:`
    (e) None of the above.

Question 15

[2.5 marks] Consider the code fragment below:

```
name = 'John Marwood Cleese'
first, middle, last = name.split()
???
```

Which of the following statements should replace the ??? above to correctly complete the code fragment and produce the following output?

```
Cleese, John Marwood
```

    (a) `print(last, first + ', ' + middle)`
    (b) `print(last + ', ' + first + ' ' + middle)`
    (c) `print(last, first middle)`
    (d) `print(last, first + ' ' + middle)`
    (e) None of the above.

Question 16
[2.5 marks] What is the output produced by the following code?

```
list1 = [10, 20, 50, 30, 80]
for element in list1:
      if element % 4 == 0:
            element = element + 6
print(list1)
```

   (a) [19, 26, 59, 39, 86]
   (b) [10, 26, 50, 36, 80]
   (c) [19, 20, 59, 39, 80]
   (d) [10, 20, 50, 30, 80]
   (e) None of the above.

Question 17
[2.5 marks] What is the output produced by the following code?

```
my_list = [12, 8, 39, 10, 28, 64]
my_list[-3] = my_list[-1] + my_list[-4]
print(my_list[-4], my_list[-3], my_list[-5], my_list[-1])
```

   (a) 39 103 8 64
   (b) 10 22 28 12
   (c) 39 36 8 64
   (d) 10 67 28 12
   (e) None of the above.

Question 18
[2.5 marks] What is the output produced by the following code?

```
list1 = [20, 40, 38, 55, 77, 99]
list2 = [33, 78, 21, 90]
list1 = list2
list2.insert(2, 60)
list2.append(99)
list1.insert(5,12)
list2.pop(4)
list1.pop()
print(list2)
```

   (a) [20, 40, 60, 38, 77, 99]
   (b) [33, 78, 60, 21, 99]
   (c) [33, 78, 60, 21, 12]
   (d) [20, 40, 55, 99, 77, 12]
   (e) None of the above.

Question 19
[2.5 marks] What is the output produced by the following code?

```
list1 = [10, 30, 50]
list2 = [20, 40, 60, 80, 100]
list1 = list1 + [70, 90]
list3 = list1[2:5:2] + list2[-1:-4:-3] * 2
print(list3)
```

   (a) [50, 70, 90, 90]
   (b) [50, 90, 100, 100]
   (c) [50, 70, 100, 40]
   (d) [70, 90, 40, 40]
   (e) None of the above.

Question 20
[2.5 marks] What is the output produced by the following code?

```
result_list = []
list1 = [10, 20, 50, 90]
list2 = [12, 36]
list2 = list1
list1[3] = 56
list2[1] = 27
result_list += [list1 == list2]
result_list += [list1 is list2]
print(result_list)
```

   (a) [True, True]
   (b) [False, True]
   (c) [True, False]
   (d) [False, False]
   (e) None of the above.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2015
Campus: City

Computer Science

Principles of Programming

(Time Allowed: TWO HOURS)

SECTION B Question/Answer Booklet

Answer all questions in this section in the space provided. If you run out of space then please use the Overflow Sheet and indicate in the allotted space that you have used the Overflow Sheet.

Surname:	
First Name(s):	
Student ID:	
Login Name (UPI):	

MARKERS ONLY

Q1 – Q20	Q23	Q26
(/50)	(/8)	(/12)
Q21	Q24	TOTAL
(/10)	(/8)	
Q22	Q25	(/100)
(/5)	(/7)	

Question 21 [10 marks]

Complete the `add_random()` function which is passed a string parameter, `word`. The function gets a random letter from the alphabet string and inserts the letter into a random position of the parameter word. Finally the function prints both the original word and the word with the added random character. For example, the following program with the completed `add_random()` function may print:

```
dog doog
dog bdog
dog dovg
```

```python
import random

def add_random(word):
```

```python
        alphabet = "abcdefghijklmnopqrstuvwxyz"
```

(10 marks)

```python
def main():
    add_random("dog")
    add_random("dog")
    add_random("dog")
main()
```

Question 22 [5 marks]

Complete the print_upside_triangle() function below which takes an integer as a parameter and prints an upside down triangle of the size specified in the parameter. For example, the following code:

```
print_upside_triangle(4)
```

will produce the output:

```
####
###
##
#
```

```
def print_upside_triangle(size):
```

(5 marks)

Question 23 [8 marks]

Complete the `print_scores()` function below which takes a list of test scores as a parameter and prints out a histogram of the scores. Each of these test scores will be an integer greater than 9 and less than 100.

The histogram should have one row for each distinct test score in the list, followed by a space, and followed by one star for each test score of that value. For example, if the tests scores in the list were 55, 80, 80, 95, 95, 95 and 98, the output to the screen may look like the following:

```
55 *
80 **
95 ***
98 *
```

Note that the order may be different from the order in the above example.

```
def print_scores(scores_list):
    dict = {}




















    for key, value in dict.items():



```
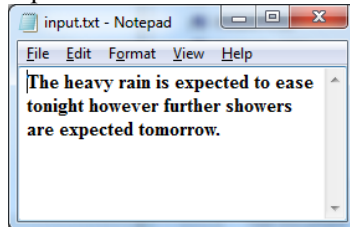
(8 marks)

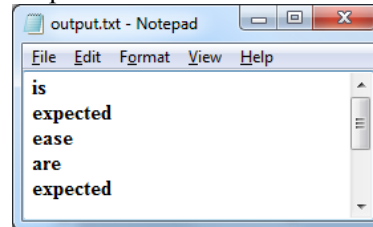## Question 24 [8 marks]

Complete the following `extract_words()` function which selects all the words in the source text file which start with a vowel and writes them one by one into a new target file (one word per line). As well as producing the new text file, the function returns the total number of words written to the output file. For example:

input.txt:


```
The heavy rain is expected to ease
tonight however further showers
are expected tomorrow.
```

output.txt


```
is
expected
ease
are
expected
```

`5 words extracted.`

```python
def main():
    total = extract_words("input.txt", "output.txt")
    print(str(total) + " words extracted.")
main()
```

```python
def extract_words(source_file, target_file):
    vowels = "aeiouAEIOU"
```

(8 marks)

Question 25 [7 marks]

Complete the following `sort_tuple()` function which sorts the elements in the tuple passed in as a parameter into descending order. The function returns the sorted tuple.

```
def sort_tuple(my_tuple):
```

(7 marks)

Question 26 [12 marks]

a) The `draw_shape()` function is passed five parameters:

   `a_canvas`: the canvas on which the shape is drawn,

   `line_is_vertical`: a boolean indicating whether the line drawn is vertical or horizontal,

   `left`: the x position of the shape (i.e., how far from the left of the canvas),

   `top`: the y position of the shape top (i.e., how far from the top of the canvas),

   `size`: the width and height of the square area covered by the shape.
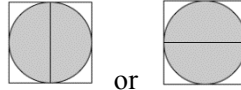
The shape drawn by the completed function is either or .

The shape is made up of an unfilled square, a circle filled with grey colour and a black line. The line which is drawn is a vertical line if the parameter, `line_is_vertical`, has the value `True`, otherwise the line drawn is a horizontal black line. The left-top position of the shape is given by the two parameters `left` and `top` and the width and height of the complete shape is given by the parameter, `size`.

Complete the definition of the `draw_shape()` function:

```
def draw_shape(a_canvas, line_is_vertical, left, top, size):
```

(6 marks)

Parts b) and c) of this question refer to the following program:

```python
from tkinter import *

def draw_shape(a_canvas, line_is_vertical, left, top, size):
    #Assume that this function has been correctly coded as
    #specified in Part a) above


# Process a row of shapes
def process_single_row(a_canvas, left, top, size):
    for num in range(3):
        draw_shape(a_canvas, True, left, top, size)
        left = left + size
        draw_shape(a_canvas, False, left, top, size)
        left = left + size


# Process some rows of shapes
def process_pattern(a_canvas, size):
    left = size
    top = size
    for num in range(2):
        process_single_row(a_canvas, left, top, size)
        top = top + size * 3
        left = left + size


def main():
    size = 10
    root = Tk()
    root.title("A Canvas")
    root.geometry("100*60+10+20")
    a_canvas = Canvas(root)
    a_canvas.pack(fill = BOTH, expand = True)
    process_pattern(a_canvas, size)
    root.mainloop()
main()
```
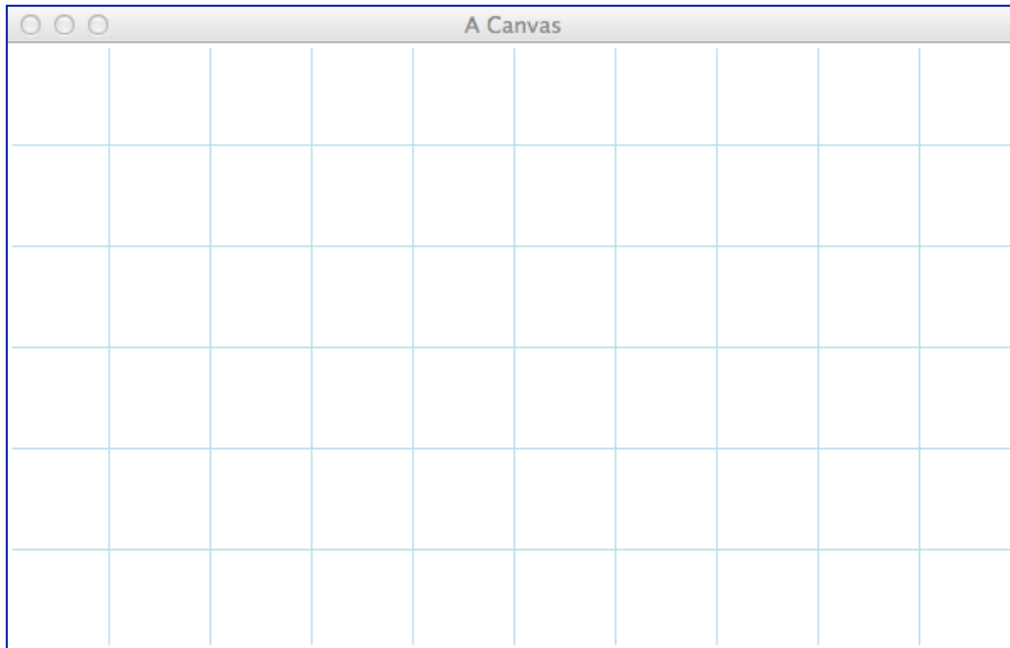
b) In total, how many complete shapes are drawn by the above program, i.e., how many times is the `draw_shape()` function called when the above program is executed?

(2 marks)

c) As accurately as possible, in the window below, show what is drawn by the above program. Grid lines have been drawn in the window to help you. The gap between adjacent gridlines is 10 pixels.

(4 marks)