THE UNIVERSITY OF AUCKLAND

Summer Semester, 2018
Campus: City

TEST
SOLUTIONS
COMPUTER SCIENCE

Principles of Programming

(Time Allowed: 75 Minutes)

NOTE:
You must answer **all** questions in this test.
No calculators or smart watches are permitted.
Answer in the space provided in this booklet.
There is space at the back for answers which overflow the allotted space.

Surname	
Forenames	
Preferred Name (if different to forenames)	
Student ID	
Username	
Lab Times	

Q1 (/20)	**Q4** (/20)
Q2 (/25)	**Q5** (/15)
Q3 (/20)	**TOTAL** (/100)

Question 1 (20 marks)

a) Complete the output produced by the following code.

```
result = 2 * 4 ** 2 + 3 - 9 // 3 ** 2
print("Result:", result)
```

```
Result: 34
```

(3 marks)

b) Complete the output produced by the following code.

```
result1 = 12 % 4 + 9 % 7
result2 = 9 // 2 + 21 / 3
print("Results:", result1, '-', result2)
```

```
Results: 2 - 11.0
```

(3 marks)

c) Complete the output produced by the following code.

```
words = "RICE CRISPS"

index1 = words.find('C')
index2 = words.rfind('C')

word1 = words[:index1]
word2 = words[index2 + 3:]
combined = word1 + word2
print("Combined:", combined)
```

```
Combined: RISPS
```

(3 marks)

d) Complete the output produced by the following code.

```
words = "reality continues to ruin my life"
result = words[8: 12] + " " + words[-2:] + " " + words[13]
print("Result:", result)
```

Result: cont fe n

(3 marks)

e) What is the smallest possible number and what is the largest possible number which can be printed by the following code?

```
import random

result1 = random.randrange(3, 10, 2)
result2 = random.randrange(5, 35, 10)
result = result1 + result2
print(result)
```

SMALLEST: 8 LARGEST: 34

(4 marks)

f) Given the following code, what is the type of each of the three Python objects: `object1`, `object2` and `object3`?

```
a_num = 345
a_list = [3, 6, 1, 4]
a_string = "456"

object1 = [a_num % 10]
object2 = a_string * a_list[0]
object3 = a_string[a_list[2]] == "6"
```

type of object1: list

type of object2: string

type of object3: boolean

(4 marks)

Question 2 (25 marks)

a) Assume that the variable, `value`, has been initialised to some integer. Write the boolean expression which tests if `value` is either a two digit positive whole number or a single digit negative number greater than -6.

```
value > 9 and value < 100 or value > -6 and
                                    value < 0
```

(4 marks)

b) Assume that the variable, `value`, has been initialised to some integer. Write the boolean expression which tests if `value` is a multiple of 3 between 5000 and 6000 (neither inclusive).

```
value % 3 == 0 and value > 5000 and value
                                    < 6000
```

(4 marks)

a) Give the output produced when the following `main()` function is executed.

```
def main():
    function_ifs(4, 6, 3)

def function_ifs(a, b, c):
    if a > b and c < b:
        print("A")
    elif not a < 5 and a > c:
        print("B")
    elif b < c or c < 5:
        print("C")
        if not a < 20:
            print("D")
    else:
        print("E")
        if a < b or c > b:
            print("F")
    if a > 4:
        print("G")
    print("H")
```

C
H

(7 marks)

b) Add code to the `main()` function below so the program does the following:

1. Using the `request` string as the prompt, prompt the user to enter a three digit code and assign the code (an integer) obtained from the user to a variable named `number`.
2. Add 2343 to the variable named `number`.
3. Concatenate the `number` from Step 2 above to the end of the string `"SPC"` and store the result in a variable named `long_code`.
4. Assign the first six characters of the `long_code` variable from Step 3 above to a variable named `code`.
5. Print the variable `code` so the output has the format `"Code: "` followed by the six `code` characters (from Step 4 above).

```
def main():
    request = "Enter your three digit code: "


    number = int(input(request))
    number = number + 2343
    long_code = 'SPC' + str(number)
    code = long_code[:6]
    print("Code:", code)
```

(10 marks)

Question 3 (20 marks)

a) Give the output produced by the following code.

```
number = 10
counter = 2

while number > 3:
    number = number - counter
    print(number, '-', counter)
    counter = counter + 1

print(number, '-', counter)
```

```
8 - 2
5 - 3
1 - 4
1 - 5
```

(6 marks)

b) Give the output produced by the following code.

```
result = 2

for number in range(1, 13, 3):
    if result % 2 == 0:
        print(number, end = " ")
    result = result + number

print(result)
```

```
1 10 24
```

(6 marks)

c) Using a `for ... in range(...)` loop, write the code which prints all the numbers between 184 and 437 (both inclusive) which are exactly divisible by 23. The numbers should all be written on a single line with a single space after each number.

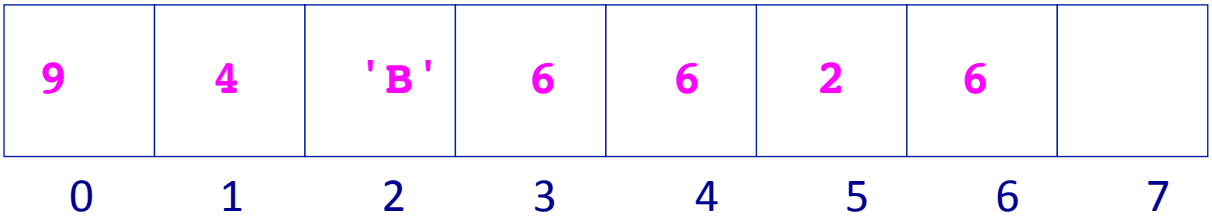
```
for number in range(184, 438):
    if number % 23 == 0:
        print(number, end = " ")
print()
```

(8 marks)

Question 4 (20 marks)

a) In the boxes below, show each element of `a_list` after the following code has been executed. Use as many of the boxes as you need.

```
a_list = ["A", 4, "B", 6, 7, 2]
a_list[0] = a_list[1] + 5
a_list[-2] = a_list[-4] * a_list[-1]
a_list[4] = a_list[5] + a_list[len(a_list[2])]
a_list = a_list                + [len(a_list)]
```



(6 marks)

b) Give the output produced by the following code.

```
number_list = [9, 3, 6, 2, 8]
jump_around = [4, -3, 5, 1]
position = 0
final_number = 0

for number in jump_around:
    position = position + number
    print(position, end = " ")
    if position > -1 and position < len(number_list):
        final_number = final_number + number_list[position]

print()
print(final_number)
```

```
4 1 6 7
11
```

(6 marks)

c) Complete the `get_list()` function which has a list of words (`word_list`) as a parameter. The function returns a new list of all the words in the parameter list which have a length which is greater than or equal to the length of the first word in the parameter list. You can assume that the parameter list always contains at least one element.

For example, executing the following `main()` function using the completed `get_list()` function gives the output:

```
['April', 'August', 'September']
```

```
def main():
    word_list = ["April", "May", "August", "July", "September"]
    words = get_list(word_list)
    print(words)

def get_list(word_list):
```

```
    list2 = []
    first_word_length = len(word_list[0])

    for word in word_list:
        if len(word) >= first_word_length:
            list2.append(word)
    return list2
```

(8 marks)

Question 5 (15 marks)

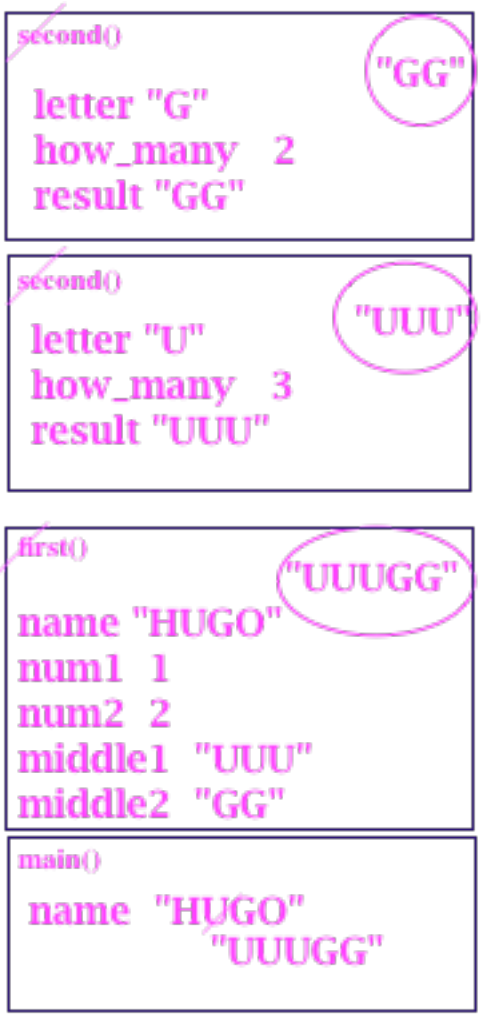
Using the code trace technique taught in lectures, perform a code trace on the following program and show the output.

```
def main():
    name = "HUGO"
    name = first(name, 1, 2)
    print("A", name)

def first(name, num1, num2):
    print("B")
    middle1 = second(name[num1], 3)
    middle2 = second(name[num2], 2)
    print("C", middle1, middle2)
    return middle1 + middle2

def second(letter, how_many):
    print("D")
    result = letter * how_many
    return result
main()
```

The output

B
D
D
C UUU GG
A UUUGG

(6 marks)



(9 marks)