THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2014
Campus: City

COMPUTER SCIENCE
TEST SOLUTIONS
Principles of Programming

(Time Allowed: 75 minutes)

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

<table>
<thead>
<tr>
<th>Surname</th>
<th>Forenames</th>
<th>Student ID</th>
<th>Login (UPI)</th>
<th>Lab Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q4</td>
<td>Q7</td>
<td>TOTAL</td>
<td></td>
</tr>
<tr>
<td>(/20)</td>
<td>(/12)</td>
<td>(/10)</td>
<td>(/100)</td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>Q5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(/21)</td>
<td>(/14)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>Q6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(/9)</td>
<td>(/14)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 1 (20 marks)

a) Complete the output produced by the following code.

```python
num1 = 4
num2 = 5
result = (num1 + num2 * 4 - num1 ** 2) / 2
print("Result:", result)
```

Result: 4.0

(2 marks)

b) Complete the output produced by the following code.

```python
num1 = 3
num2 = 5
result = str(num1 + num2 * num1 // 2) + "00"
print("Result:", result)
```

Result: 1000

(2 marks)

c) Give the largest number and the smallest number which can be printed by the following code.

```python
number = random.randrange(-5, 20, 5)
print(number)
```

Largest number: 15
Smallest number: -5

(2 marks)
d) Complete the output produced by the following code.

```python
num1 = 7
num2 = 5
result = num1 % num2 + num2 % num1
print("Result:", result)
```

Result: 7

(2 marks)

e) Complete the output produced by the following code.

```python
num1 = 15
num2 = 4
result = (num1 // num2) + num1 / (num2 - 2)
print("Result:", result)
```

Result: 10.5

(2 marks)
f) Given the following function definition:

```python
def print_result(num1, num2, message):
    portion = message[min(num1, num2) : max(num1, num2)]
    print("*" + portion + "*")
```

what is the output produced by the following function call?

```python
print_result(8, 3, "Tricky Question")
```

```
cky Q*
```

(2 marks)

g) Given the following function definition:

```python
def get_number(prompt):
    user_input = input(prompt)
    user_input = user_input[0] + "0" * (len(user_input) - 1)
    return user_input
```

complete the output produced when the user enters 87692 at the prompt.

```python
number = get_number("Enter a number: ")
print("Number: ", number)
```

```
Enter a number: 87692
Number: 80000
```

(2 marks)
h) Complete the output produced by the following code.

```python
phrase = "Sing out loud"
pos1 = phrase.find("ou")
pos2 = phrase.rfind("ou")
print("Result:", pos1, pos2)
```

```
Result: 5 10
```

(2 marks)

i) The following code prompts the user for their year of birth, gets the user's year of birth and prints the user's age this year. Complete the code.

```python
prompt = "Enter year of birth: "
birth_year = input(prompt)
age = 2014 - int(birth_year)
print("Age this year:", age)
```

(2 marks)

j) Complete the following statement which assigns a list containing the following ten integer values: 10, 20, 30, ..., 100 in that order, to the variable `int_list`.

```python
int_list = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

(2 marks)
**Question 2 (21 marks)**

a) Complete the `adjust_word()` function which is passed two parameters: a word and a number (the required length of the string to be returned).

The function returns a string, the length of which is given by the parameter, `length`.

- If the parameter, `word`, has the same length as the required length, then the word is returned unchanged.
- If the parameter, `word`, is longer than the required length, the word is truncated to the required number of letters.
- If the word is shorter than the required length, the word is lengthened by adding the correct number of stars.

For example the following code:

```python
word1 = adjust_word("cats", 4)
word2 = adjust_word("dogs", 7)
word3 = adjust_word("elephant", 5)
print(word1, word2, word3)
```

prints:
```
cats dogs*** eleph
```

```python
def adjust_word(word, length):
    extras = "*" * length
    word = word + extras
    return word[0: length]
```

(7 marks)

b) Complete the `fiddle()` function which is passed two words as parameters. The function swaps the first character of the two words and returns the phrase made up of the two words separated by a blank space. For example, the following code:

```python
print(fiddle("eat", "bats"))
print(fiddle("bun", "fight"))
```

prints:
```
bat eats
fun bight
```
def fiddle(word1, word2):
    letter1 = word1[0]
    letter2 = word2[0]

    word1 = letter2 + word1[1:]
    word2 = letter1 + word2[1:]

    return word1 + " " + word2

(7 marks)

c) Complete the get_cost_message() function which is passed three parameters: the number of items, the cost of each item and the percent discount. The function works out the total cost of the items minus the discount and returns a string made up of the string, "Number of items ", followed by the number of items, followed by ": $" and finally the cost. For example, the following code:

    message1 = get_cost_message(2, 12, 10)
    message2 = get_cost_message(20, 6, 20)
    print(message1)
    print(message2)

prints:
Number of items 2: $21.6
Number of items 20: $96.0

def get_cost_message(number_of_items, cost_each, discount_percent):
    cost = number_of_items * cost_each
    discount = cost * discount_percent / 100
    to_pay = cost - discount

(7 marks)
message = "Number of items " + 
str(number_of_items) + 
": "$ + str(to_pay)

return message

(7 marks)
Question 3 (9 marks)

a) Assume that letter is a variable which has been assigned a string value. Write a boolean expression whose value is True if and only if letter is a newline character.

```
letter == "\n"
```

(3 marks)

b) Assume that a variable hours_worked has been initialised. Complete the assignment statement which assigns the value True to the variable worked_overtime if hours_worked is greater than 40 and False otherwise.

```
worked_overtime = hours_worked > 40
```

(3 marks)

c) Given the already defined variables temperature and humidity, write a boolean expression which evaluates to True if the temperature is greater than 90 and the humidity is less than 10.

```
temperature > 90 and humidity < 10
```

(3 marks)
Question 4 (12 marks)

a) Write an if statement which multiplies the value associated with the variable, pay, by one-and-a-half if worked_overtime is True.

```python
if worked_overtime:
    pay = pay * 1.5
```

(3 marks)

b) Write an if/else statement which assigns True to the variable fever if temperature is greater than 98.6, otherwise it assigns False to fever.

```python
if temperature > 98.6:
    fever = True
else:
    fever = False
```

(3 marks)

c) Complete the compare_ints() function using an if/elif statement. The function compares the two integer values given by the parameters first and second and returns 0 if the two numbers are equal in value, 1 if the first number is greater than the second one, and -1 otherwise.

```python
def compare_ints(first, second):
    if first == second:
        return 0
    elif first > second:
        return 1
    else:
        return -1
```

(6 marks)
Question 5 (14 marks)

a) Use a while loop to complete the `sum_squares()` function, which computes the sum of the squares of the numbers from 1 up to (and including) the value given by the parameter, `number`. The function returns the sum. Note that you **MUST** use a `while` loop to solve the problem.

```python
def sum_squares(number):
    count = 0
    sum = 0
    while count <= number:
        count += 1
        sum += count * count
    return sum
```

(7 marks)

b) Rewrite the `sum_squares()` function (question 5 part a) above, using a `for` loop structure.

```python
def sum_squares(number):
    sum = 0
    for i in range(1, number+1):
        sum = sum + i * i
    return sum
```

(7 marks)
Question 6 (14 marks)

a) Complete the `get_duplicate_list()` function, which copies the content of the parameter list, `source`, into a new list. The function returns the new list which has the exact same content as the parameter list:

```python
def get_duplicate_list(source):
    target = []
    for element in source:
        target = target + [element]
    return target
```

(7 marks)

b) Complete the `get_index_of_largest()` function which finds the largest value inside the parameter list of integers, `numbers`. The function returns the index of the largest number in the list. You can assume that all the number in the list are unique.

```python
def get_index_of_largest(numbers):
    index = 0
    for i in range(1, len(numbers)):
        if numbers[index] < numbers[i]:
            index = i
    return index
```

(7 marks)
**Question 7 (10 marks)**

a) Perform a code trace for the following program and give the output. Give the output in the space below and show the code trace in the space provided on the next page.

```python
def main():
    number = 4975
    length = 4
    result1 = function1(number, 0)
    print("A")
    result2 = function1(number, length - 1)
    print("B")
    result = result1 + result2
    print("C", result)

def function1(num1, num2):
    num1 = str(num1)
    result = num1[num2]
    print("D", result)
    return result

main()
```

Give the output:

```
D 4
A
D 5
B
C 45
```
Show the code trace in the space below:

```plaintext
function1()
num1 "4975"
num2 3
result "5"

function1()
num1 "4975"
num2 0
result "4"

main()
number 4975
length 4
result1 "4"
result2 "5"
result "45"
```

(4 marks)