THE AUSTRALIAN NATIONAL UNIVERSITY

Second Semester 2007 — Final Examination

COMP2720
Automating Tools for New Media

Writing Period: 180 minutes

Study Period: 15 minutes

Permitted Materials: One A4 page with notes on both sides.
NO calculator permitted.

Answer all questions

Write your answers in the boxes provided in this booklet. Only those answers written in this booklet will be marked. There is additional space at the end of the booklet in case the boxes provided are insufficient. Label any answers you write at the end of the booklet to indicate which question they refer to, and write "continued on page…" at the point where you run out of space. Do not remove this booklet from the examination room. ***Do not write in red ink anywhere in the booklet. Please write clearly — if we cannot read your writing you might lose marks!***

Name (family name first):

Student Number:

Official use only:

Q1 (6)	Q2 (11)	Q3 (5)	Q4 (9)	Q5 (9)	Total (40)

QUESTION 1 [6 marks]

(a) Explain what an *analogue* signal is. Name two communication media, which used to be analogue, but which have recently been, or are currently being converted into the digital format worldwide (including Australia).

QUESTION 1(a)	**[1 mark]**

(b) If signals (and media in general) were previously analogue, does this mean that they could not be handled by computer? If computers *were* used, were they different from modern digital devices?

QUESTION 1(b)	**[1 mark]**

(c) What is *Moore's Law*? What it has got to do with the advances in digital media processing?

QUESTION 1(c)	**[1 mark]**

(d) Explain how text media can be digitized. Is it crucial what language the text is written in?

QUESTION 1(d)	**[1 mark]**

(e) Explain why is it useful to study media processing as *computation* in contrast to simply learning how to use a media processing tool (like Adobe's *Photoshop*^{TM} for image processing, or Apple's *Final Cut Pro*^{TM} for movies).

QUESTION 1(e)	**[2 marks]**

QUESTION 2 [11 marks]

(a) The colour encoding scheme *YIQ* was used in the past in the USA commercial television broadcasting. The relationship between the primaries in *RGB* and *YIQ* is as follows:

$$\begin{aligned} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ I &= 0.596 \times R - 0.275 \times G - 0.321 \times B \\ Q &= 0.212 \times R - 0.523 \times G + 0.311 \times B \end{aligned}$$

It is the values (Y, I, Q) for every pixel which were broadcasted. To optimally use the limited bandwidth, the Y component represented luminance, and was encoded with *more* bits to ensure higher accuracy. Meanwhile the chromaticity was encoded in I and Q, and they each used *fewer* bits since colour perception is less accurate then luminance.

Write down a *Python/JES* function `rgb2yiq(pic)` that takes a picture input `pic`, and replaces every pixel's colour components with the new values as if they were intended for TV broadcasting in the *YIQ* scheme (we can imaging that after receiving the signal the *YIQ* values are converted back into the *RGB* values and the whole picture is displayed on the TV screen, **but your `rgb2yiq(pic)` needs not to do this**).

QUESTION 2(a)	**[3 marks]**

(b) A recent publication in the *Nature Physics* magazine (vol. **3**, Oct. 2007, p. 685) reports that the *very high resolution images* of the Moon surface, which were taken during the last manned mission of *Apollo 17* in 1972, have been scanned and converted into digital format. These images, available from the Arizona State University web site at `http://apollo.sese.asu.edu`, are *very large* — an average file is 1.3 gigabytes. The images have the resolution of *200 pixels per millimeter*, and every pixel is encoded as a shade of gray using 14 bits!

Calculate the *geometric* size of such image assuming it has 1:2 aspect ratio (the ratio of height to width). ***Hint:*** Use the approximation $2^{10} = 1024 \approx 1,000$.

QUESTION 2(b)	**[2 marks]**

(c) Explain why some popular image formats (like *JPEG*), which involve *lossy* compression, are nevertheless widely used to store images of high quality.

QUESTION 2(c)	**[1 mark]**

(d) The *run length encoding (RLE)* is a compression technique used in digital media and signal transmission. What is the essence of this technique? Is it lossy or lossless?

QUESTION 2(d)	**[2 marks]**

(e) Explain in one sentence what the following *Python/JES* function is doing to the image which is passed as parameter `pic`. Use a sketch if it helps.

```
def mischievous(pic):
  w = getWidth(pic)
  h = getHeight(pic)
    for y in range(1,h/2 + 1):
      for x in range(1,w/2 + 1):
        lu = getPixel(pic,x,y)
        ru = getPixel(pic,w-x+1,y)
        lb = getPixel(pic,x,h-y+1)
        rb = getPixel(pic,w-x+1,h-y+1)
        c1 = getColor(lu)
        if (x >= y):
          c2 = getColor(ru)
          setColor(lu,getColor(lb))
          setColor(ru,getColor(rb))
          setColor(lb,c1)
          setColor(rb,c2)
        else:
          c2 = getColor(lb)
          setColor(lu,getColor(ru))
          setColor(lb,getColor(rb))
          setColor(ru,c1)
          setColor(rb,c2)
```

QUESTION 2(e)	**[3 marks]**

QUESTION 3 [5 marks]

(a) What units are used to measure the intensity of sound? What corresponds to the zero value of the unit for the sound heard by the human ear?

QUESTION 3(a)	[1 mark]

(b) The *Nyquist theorem* relates the number of samples per second (*sampling rate*) N_{sr} used to digitally encode a sound, and the maximal frequency F_{max} which can be captured by this encoding. The theorem says that

$$N_{sr} \approx 2 \times F_{max}.$$

Explain why the relationship contains the coefficient 2, but does not simply requires $N_{sr} \approx F_{max}$? Use a waveform sketch if it helps you.

QUESTION 3(b)	[2 marks]

(c) Consider the following *Python/JES* function:

```
def soundfunction(sound):
  factor = 0
  length = getLength(sound)
  inc = 2.0 / length
  i = 0
  for sample in getSamples(sound):
    value = getSample(sample)
    if (i < length / 2):
      factor = factor + inc
      setSample(sample,value * factor)
    else:
      factor = factor - inc
      setSample(sample,value * factor)
    i = i+1
```

Explain briefly what this function does to the original `sound` when it is passed as a parameter to `soundfunction(sound)`.

QUESTION 3(c)	[1 mark]

(d) The `normalise(sound)` function was used in the Laboratory 3 to uniformly (without distorting) amplify the `sound` to the maximum possible volume and avoid *clipping*. Explain how the number of bytes used to encode every sample (1, 2, or more) affects the amount by which every sample is amplified.

QUESTION 3(d)	[1 mark]

QUESTION 4 [9 marks]

(a) Write a Python function `countcwl(fname)` which will open a text file `'fname'`, and return the number of *characters, words and lines* this file contains (the return value must be a tuple of three integers). Assume that all characters in `'fname'` are *ASCII* code.

QUESTION 4(a) **[3 marks]**

(b) What are the three types of quoting into which a Python string can be enclosed? Give examples.

QUESTION 4(b) **[1 mark]**

(c) Assume that you have the following sequence of Python instructions. First, a list with 15 random single-digit integers is generated (never mind how):

```
>>> ilist = [6, 1, 6, 1, 8, 2, 4, 1, 0, 4, 7, 8, 4, 1, 9]
```

Then several statements are executed, (i)–(iv). Write down what will be printed as the result of executing each of the statements:

(i) `>>> print ilist[2] % ilist[-3]`

QUESTION 4(c)[i] **[1 mark]**

(ii) `>>> print ilist[2:].count(8)`

QUESTION 4(c)[ii] **[1 mark]**

(iii)
```
>>> newlist = ilist[10::-3]
>>> newlist.sort()
>>> print newlist
```

QUESTION 4(c)[iii] **[1 mark]**

(iv) `>>> print sum(ilist[1:13:2])`

QUESTION 4(c)[iv] **[1 mark]**

(d) What Python *standard library modules* did you use in the *extra work* which you have done for the Comp2720 Portfolio, and why their use was necessary. If you did not do the extra work, tell what standard library modules you used for the Assignment 2 program.

QUESTION 4(d) **[1 mark]**

QUESTION 5 [9 marks]

(a) *Jython* is an implementation of the *Python* language which is based on the *Java programming environment.* Explain what benefits this implementation has, and what price one has to pay for them.

QUESTION 5(a)	**[1 mark]**

(b) To what type of *linearly organized data* (like array, list or dictionary) can the *binary search algorithm* be applied. If the size of the data is N, what is the performance of the binary search algorithm in terms of $\mathcal{O}(\,)$ notations.

QUESTION 5(b)	**[1 mark]**

(c) The $P = NP$ problem is one of the most important *unsolved* problems in Computer Science. Explain how you understand the $P = NP$ problem. Why is it so important?

QUESTION 5(c)	**[1 mark]**

(d) Below is a recursively defined function `fib(n)`, which takes a non-negative integer number n as an input, and returns the list of the first n *Fibonacci numbers*, F_n (elements of the *Fibonacci sequence*):

```
def fib(n):
   if n < 1:
      return [] # if input is not positive an empty list is returned
   elif n == 1:
      return [1]
   elif n == 2:
      return [1,1]
   else:
      list = fib(n-1)[:]
      list.append(list[n-2] + list[n-3])
      return list
```

When called, the function behaves as follows:

```
>>> fib(15)
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]
```

The ratio of two successive Fibonacci numbers, $\varphi_n = F_{n+1}/F_n$, represents a fraction which tends to the *Golden Mean* number, φ, when the value of n is becoming very large. The numbers φ_n provide the so called *rational approximations* to the Golden Mean φ.

Write a function `gm(n)`, which takes a non-negative integer number n as an input, and returns the number φ_n. This function has to call `fib(n)` function, it has to be written in the *functional programming style*, using *lambda notation* and the *meta-function* `map`. (***Hint:*** All numbers F_n are integers, so be careful about the integer division when calculating the ratio F_{n+1}/F_n.)

QUESTION 5(d)	**[3 marks]**

(e) One advantage of using functions is that they allow efficient code reuse. Give two (2) other advantages of using functions.

QUESTION 5(e)	**[1 mark]**

(f) Below is a Python definition of the class *Weather*:

```
class Weather:
   def __init__(self, d, p, t):
      self.day = d
      self.precis = p
      self.temp = t

   def show_weather(self):
      w = "Forecast for "
      w += self.day + ": "
      w += self.precis + ", "
      w += str(self.temp)
      print w
```

How do you create an object w of the *Weather* classs which corresponds to "Monday" (as 'day'), "Rain" (as 'precis') and 27 (as 'temp'), and then print the weather forecast using w?

QUESTION 5(f)	**[1 mark]**

(g) So, why do you like Python?

QUESTION 5(g)	**[1 mark]**

Additional answers to QUESTION __(__)

Additional answers to QUESTION __(__)

Additional answers to QUESTION __(__)