TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2016
TRIMESTER 1

NWEN 241

SYSTEMS PROGRAMMING

Time allowed: TWO HOURS

CLOSED BOOK

Permitted materials: No calculators are allowed.

No electronic dictionaries are allowed.

Paper foreign to English language dictionaries are allowed.

Instructions: The examination contains 5 questions. You must answer ALL questions.

The exam consists of 100 marks in total, with 20 marks for each of the 5 questions:

Question 1 C General Questions	[20 marks]
Question 2 Arrays, Pointers and File Handling	[20 marks]
Question 3 Bitwise Operators and Data Structures	[20 marks]
Question 4 Python Fundamentals	[20 marks]
Question 5 Writing and Reading Python Programs	[20 marks]

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Student ID: .....

Question 1. C General Questions **[20 marks]**

(a) [4 Marks] Explain the four steps of compilation for C programs.

(b) [6 Marks] Explain how the Stack, Heap and Data Segment sections are used in program memory and how these sections relate to compile-time or run-time memory allocation.

(c) [6 Marks] Explain the difference between Java Class objects and C Structure variables.

(d) [4 Marks] Discuss pass-by-value and pass-by-reference in C and Java.

Question 2. Arrays, Pointers and File Handling **[20 marks]**

(a) [2 Marks] Write a `scanf` statement that takes all characters except the new-line character.

(b) [4 Marks] Give a declaration for the variable `p` in each of the following cases.

`p` is a pointer to an element of a string.

`p` is an array of n pointers to `char`.

`p` is a function that takes no arguments and returns a pointer to `int`.

`p` is a pointer to a function that takes two arguments: a pointer to `int` and a pointer to a pointer to `int` and returns a pointer to an array of n pointers to `char`.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

(c) [8 Marks] Consider the following code:

```
char *a[] = {"AAA", "BBB", "CCC"};

// for you to complete - n is the number of elements in a
int n = ...;

// for you to complete - declare ptr
...
ptr = &a;

printStr(ptr,n);
```

Define the `int` variable `n` , declare variable ptr, and implement function `printStr` so that `printStr(ptr,n)` prints out the three strings in the following format:

AAA
BBB
CCC

(d) [6 Marks] Write a command-line-arguments based program. The program will be called with two file names as its command line arguments. The program should read the characters from the first file, delete any character which is an odd number (1, 3, 5, 7, 9), and then write it to the second file. You **must use pointer notation** to implement this program.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 3. Bitwise Operators and Data Structures **[20 marks]**

(a) [10 Marks] In the following, we have defined a structure type named charNode:

```
#define node_size sizeof(charNode)
typedef struct charNode charNode;
typedef charNode *ptr_charNode;

struct charNode {
  char data;
  ptr_charNode next;
};
```

Write a function with prototype `ptr_charNode charList(char *)`, which creates a `charNode` node for each character in a string and links the nodes in sequence, and returns a pointer to the first node of the resulting list.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

(b) [10 Marks] See the program below.

```
int main(void)
{
  int age[] = {8, 2, 6, ...,12, 15, 11};
  ...
  int *ageInt = calloc(m, sizeof(int));
  ...
  return 0;
}
```

In the box below, complete this program. Assume that the ages in array `age` are between 0 ~ 15. Pack all the ages into the memory space allocated to **`ageInt`**. You need minimise the memory space required for packing, that is, you need minimise the value of **`m`**. The ages need to be packed in order from high-order bits to low-order bits in the memory space. Assume that you are working on a 32-bit machine where the **`sizeof(int)`** is **4 bytes**.

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 4. Python Fundamentals **[20 marks]**

a) [5 marks] Some of following strings are legal Python identifiers and some are not. For each one state if the name is legal and if not, explain why:

i) `total_time`

ii) `pass`

iii) `2nd_day`

iv) `numberOfVists`

v) `user-count`

b) [5 marks] Consider the following python code:

```
#!/usr/bin/env python3

firstList = [5, 15, 2, 22]
firstList.sort()
firstList.append(20)
secondList = firstList[2:]
```

What would the output be from the following python commands?

i) `print(firstList)`

ii) `print(len(secondList))`

iii) `print(firstList == secondList)`

iv) `print(firstList[-1])`

v) `print(firstList + secondList)`

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

c) [10 marks] For each of the following Python keywords, briefly explain what it does and write a short piece of code to illustrate its use.

i) `elif`

ii) `with`

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 5. Writing and Reading Python Programs **[20 marks]**

a) [10 marks] Write a Python program that reads two strings as parameters on the command line and prints out the longer of the two. If they are the same length, print both. For example, it might be called by:

```
$ python3 longeststring.py first second
```

Your program should check the parameters and issue appropriate error messages and return values.

b) [10 marks] Consider the following python program which retrieves earthquake data from the geonet website for various periods of time. You are to add 10 suitable comments (one comment for each of the marking boxes) to the following code explaining the functionality.

Additional information

Data returned from the fp.read() statement in the code is a sequence of bytes that are encoded in the UTF-8 character set. The returned data is a well-formed series of Comma Separated Values (CSV). Each line in the CSV refers to a detected earthquake. The 9th value of each line is the magnitude of the earthquake.

```
#!/usr/bin/env python3

import sys, os
import datetime
import urllib.request

months = ['January', 'February', 'March', 'April', 'May', 'June',
'July', 'August', 'September', 'October', 'November', 'December']

urlbase = "http://wfs.geonet.org.nz/geonet/ows?output=csv&"
```

```
# (1)
#
#
#

thisyear = datetime.datetime.now().year
```

```
earthquakes = {}

for month in months:
```

```
  # (2)
  #
  #
  #
  start_date = datetime.datetime.strptime('%s-%s' % (thisyear,
month), '%Y-%B')
  end_date = start_date + datetime.timedelta(days=30)
```

```
  # (3)
  #
  #
  #
  url = urlbase + 'from=' + start_date.strftime("'%Y-%m-%d'") +
'&to=' + end_date.strftime("'%Y-%m-%d'")
```

```
  # (4)
  #
  #
  #
  try:
    fp = urllib.request.urlopen(url)
    mybytes = fp.read()
    fp.close()
  except:
    break
```

```
  end_date = start_date + datetime.timedelta(days=30)
```

```
  # (5)
  #
  #
  #
  mycsv = str(mybytes.decode('utf-8'))
```

```
  events = []
```

```
    # (6)
    #
    #
    #
    lines = mycsv.split('\n')
    for line in lines:
```

```
      try:
```

```
        # (7)
        #
        #
        #
        values = line.split(',')
        magnitude = float(values[8])
        events.append(magnitude)
```

```
      except:
        pass
```

```
  # (8)
  #
  #
  #
  sorted_events = sorted(events, reverse=True)
```

```
  # (9)
  #
  #
  #
  try:
    results = {month: {'max' : sorted_events[0], 'avg' :
sum(sorted_events)/len(sorted_events)}}ss
    earthquakes.update(results)
  except:
    break
```

```
# (10)
#
#
#
for month, results in earthquakes.items():
    print (month + " " + repr(results))
```

Student ID: .....

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

* * * * * * * * * * * * * * * * * * * *