



THE UNIVERSITY OF
**WESTERN
AUSTRALIA**

Computer Science and Software Engineering

SEMESTER 2, 2017 FINAL EXAMINATIONS

**CITS2002
Systems Programming**

FAMILY NAME: _____ GIVEN NAMES: _____

STUDENT ID:

--	--	--	--	--	--	--	--

 SIGNATURE: _____

This Paper Contains: **6 pages (including title page)**
Time allowed: **2:00 hours (including reading time)**

INSTRUCTIONS:

Answer all 6 questions. Each question is worth 10 marks.

PLEASE NOTE

Examination candidates may only bring authorised materials into the examination room. If a supervisor finds, during the examination, that you have unauthorised material, in whatever form, in the vicinity of your desk or on your person, whether in the examination room or the toilets or en route to/from the toilets, the matter will be reported to the head of school and disciplinary action will normally be taken against you. This action may result in your being deprived of any credit for this examination or even, in some cases, for the whole unit. This will apply regardless of whether the material has been used at the time it is found.

Therefore, any candidate who has brought any unauthorised material whatsoever into the examination room should declare it to the supervisor immediately. Candidates who are uncertain whether any material is authorised should ask the supervisor for clarification.

Supervisors Only - Student left at:

This page has been left intentionally blank

- 1) Consider the function `dayOfWeek()`, whose C99 prototype follows:

```
int dayOfWeek(int day, int month, int year, int firstJan);
```

The function takes four integer parameters, each providing information about a date and its year. The goal of the function is to firstly ensure that the provided date is valid and, if so, to determine on which day of the week the date falls.

The first parameter, `day`, provides the day of interest – its values must range from 1 to 31 (inclusive). The second parameter, `month`, provides the month of interest – its values must range from 1 to 12 (inclusive). The third parameter, `year`, provides the year of interest – any integer value of 1970 or greater is permitted.

The final parameter, `firstJan`, indicates the day of the week on which the first of January falls in the provided year. Its permitted values are 0 (representing Sunday), 1 (representing Monday), and so on, up to 6 (representing Saturday).

If all provided parameters are valid, the function will return the day of the week on which the indicated date falls. For example, the call

```
dayOfWeek(13, 11, 2017, 0);
```

will return the integer 1 (representing Monday).

If the provided date is invalid, the function should return the integer -1.

Write the function `dayOfWeek ()` in C99.

(10)

- 2) A *concordance* is a list of words appearing in a body of text. The list of words includes only "true" words, consisting entirely of alphabetic characters, after all punctuation and whitespace characters have been discarded.

Write a C99 function, with the prototype:

```
char **concordance(char *textfilename, int *nwords);
```

to build a concordance from a named text file.

The first parameter, `textfilename`, provides the name of the text file containing the words. You may assume that the text file contains only alphabetic, punctuation, and whitespace characters.

The second parameter, `nwords`, provides a pointer to an integer. On successful return from the function, the integer pointed to by `nwords` will contain the number of distinct words found in the file.

On successful return, `concordance` will return a vector of strings, containing the distinct words found in the text file.

If any problems are detected during the execution of `concordance`, the function should return the `NULL` pointer.

(10)

3a) Explain clearly the differences between *paging* and *segmentation* of memory.
What are the advantages and disadvantages of each scheme?
(5)

3b) With the aid of diagrams, explain the differences between *internal* memory fragmentation and *external* memory fragmentation.
(5)

4a) With reference to two distinct examples, explain *The Principle of Referential Locality*.
(5)

4b) Explain how an operating system's use of virtual memory can enable the operating system to appear to support the use of more memory than is physically installed in a computer.
(5)

5a) With reference to a diagram, explain the actions of a Unix-based operating system when a process invokes the `fork ()` system-call.
(5)

5b) Three common file allocation mechanisms, in order of increasing complexity, are *contiguous*, *chaining*, and *indexed*.

Briefly describe one advantage and one disadvantage of each mechanism.
(5)

- 6) Consider a hierarchical file-system consisting of directories, each of which possibly contains files and other sub-directories. There are no other types of file-system objects in this file-system.

When wishing to perform an incremental backup of all files in and below a named directory of the file-system, we require knowledge of how much space will be required to copy all files.

However, for an incremental backup, we don't wish to copy all files, but only the files that have been modified since a certain time.

Write a C99 function to return the number of bytes contained in all files below a named directory that have been modified since an indicated time.

Your function should have the following prototype:

```
int totalBytes(char *dirname, time_t since);
```

Any errors resulting from conditions such as being unable to access a sub-directory or a file should simply be ignored.

If `dirname` can not be opened as a directory, your function should return -1.

(10)

END OF PAPER