THE UNIVERSITY OF MELBOURNE
DEPARTMENT OF COMPUTING AND INFORMATION SYSTEMS

SAMPLE FINAL EXAM ONE – SEMESTER 2, 2016
COMP20005 ENGINEERING COMPUTATION

Student ID:

Reading Time: **fifteen** minutes
Writing Time: **two** hours

Total marks for this Exam: 60
This exam has 4 pages.

Identical Examination Papers: None
Common Content Papers: None

Authorised Materials:
Writing materials, e.g., pens, pencils, are allowed.
Books, calculators, and dictionaries are *not* allowed.

Instructions to Invigilators:
Supply students with standard script book(s).

The exam paper must remain in the exam room and be returned to the subject coordinator.

Instructions to Students:

- Attempt all questions.
- You may attempt the questions in any order. *However, you should write your answers that belong to the same question together.*
- Clearly write your answers. Any unreadable answer will be considered wrong.
- You are not required to write comments in any of your code fragments or functions. If a question says “write a function”, you may write appropriate further functions if you believe that a decomposition of the problem is appropriate.
- You may make use of library functions except when their use is explicitly prohibited. If you do make use of library functions, you must add suitable `#include` lines at the start of each corresponding answer.
- Constants should be `#define`’d prior to functions, when appropriate.

1. Short answer questions **[3 marks for each question]**

(1) In a 16-bit two's complement number representation for integers, what bit pattern represents the decimal number 123, and what bit pattern represents the decimal number -123?

(2) Assume a 16-bit floating point number system where the most significant bit represents the sign; the following 3 bits represent an integer exponent of 2 with two's complement representation; and the rest of the bits represent the mantissa. For example, decimal number 0.375 is represented by 0 111 1100 0000 0000 in this system. Then for decimal number 0.625, what will it be represented by in this system?

(3) State two desired properties of numeric processing algorithms (among those listed in the lecture slides) that are different from the desired properties of symbolic processing algorithms.

(4) In general, to fit `n` arbitrary points, a polynomial function of degree `x` is needed. Here, what would be a possible value of `x`?

(5) State the conditions required for applying the "generate and test" technique.

Programming questions

2. **[10 marks]** Write a function `void reverseArray(int intArray[], int n)` that reverses the order of `n` integers in `intArray`.

For example, if `intArray = {1, 3, 8, 6, 2}` and `n = 5` is given to the function, then after calling the function, `intArray` should become `{2, 6, 8, 3, 1}`.

You may assume that `intArray` contains at least one integer. You may **NOT** define any new arrays in the `reverseArray` function (that is, you code must operate on `intArray` itself only).

3. **[10 marks]** Write a function `void myStrCat(char *dst, char *src)` that append string `src` to the end of string `dst`.

For example, if `dst = "abc"` and `src = "def"`, then the call `myStrCat(dst, src)` will change `dst` to `"abcdef"`.

You may assume that both `dst` and `src` contain at least one character, and that `dst` has sufficient space to store the new characters from `src`. You may **NOT** change `src` or make use of any functions in the `<string.h>` library.

4. [15 marks] A rectangle is represented by its lower and upper bounds in the x-dimension and lower and upper bounds in the y-dimension, denoted by `lx, ux, ly, uy,` respectively. These bounds should be stored by `double` variables.

(1) Write the definition of a `struct` type named `rectangle_t` that represents a rectangle following the description above.

(2) Write a function `int intersect(rectangle_t rect1, rectangle_t rect2)` that returns `1` if the two rectangles `rect1` and `rect2` given interest each other. The function should return `0` otherwise.

To check whether `rect1` and `rect2` intersect, your function `intersect()` needs to check whether the bounds of the two rectangles overlap with each other in both dimensions. Note that if the two rectangles only overlap at a vertex or an edge, they are still considered intersected.

(3) Write a function `int countIntersect(rectangle_t rects1[], int n1, rectangle_t rects2[], int n2)` that counts and returns how many pairs of rectangles from the two arrays `rects1` and `rects2` intersect each other. Here, `n1` and `n2` are the size of the two arrays, respectively.

You may assume that `n1 >= 0` and `n2 >= 0`.

5. [5 marks] Write a ***recursive*** function `int isPalindrome(char *str, int n)` that returns `1` if `str` is a palindrome, that is, reads exactly the same forwards as well as backwards. If `str` is not a palindrome, then the function should return `0`. Here, `n` is the length of the string `str`.

For example, if `str = “rats live on no evil star”`, then the call `isPalindrome(str, 25)` should return `1`. If `str = “abab”`, then the call `isPalindrome(str, 4)` should return `0`.

If you use iteration rather than recursion to answer this question, the full mark of this question will reduce to **2 marks**.

You may assume that `str` contains lowercase English characters only.

6. **[5 marks]** Write a function `int randomisedSubsetSum(int items[], int n, int k)` that uses a randomised strategy to solve the subset sum problem with the set of `n` `(0 < n < 100)` items represented by the `items` array, and the sum to achieve represented by `k`.

This randomised strategy repeats the following steps for `10,000` iterations. At each iteration, it randomly chooses an integer `num (0 < num <= n)`. Then it randomly chooses `num` items from the items array. If these `num` items add to `k`, then the function `randomisedSubsetSum()` returns `1`. Otherwise, it starts the next iteration. When `10,000` iterations are completed, the function `randomisedSubsetSum` returns `0`.

When choosing the `num` items randomly, you need to find a way to guarantee that no item is chosen twice from the `items` array.

You need to add suitable `#include` lines if you use any library functions.

End of exam