THE UNIVERSITY OF MELBOURNE
DEPARTMENT OF COMPUTING AND INFORMATION SYSTEMS

ANOTHER MID-SEMESTER TEST SAMPLE ANSWER – SEMESTER 2, 2016
COMP20005 ENGINEERING COMPUTATION

Total marks for this Exam: 10

Reading Time: **5** minutes
Writing Time: **30** minutes

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 Students:

- Attempt all questions.
- Clearly write your answers. Any unreadable answer will be considered wrong.

1. **[3 marks]** Below is a program that reads in a piece of text, removes the lines starting with character ‘C’, and then outputs the rest of the text. Write the guards to be used in lines 5, 8, and 11 to realise the functionality of the program.

```
0:      #include <stdio.h>
1:      int
2:      main(int argc, char *argv[]) {
3:              int ch, incomment = 0, newlinelast = 1;
4:              while ((ch = getchar()) != EOF) {
5:                      if (/* add guard here */) {
6:                              incomment = 1;
7:                      }
8:                      if (/* add guard here */) {
9:                              putchar(ch);
10:                     }
11:                     if (/* add guard here */) {
12:                             incomment = 0;
13:                             newlinelast = 1;
14:                     } else {
15:                             newlinelast = 0;
16:                     }
17:             }
18:             return 0;
19:     }
```

Line 5:	ch == ‘C’ && newlinelast == 1
Line 8:	incomment == 0
Line 11:	ch == ‘\n’

2. [4 marks] Complete the following program to read English letters, and check if they are in non-increasing order. Any letter that is larger than the one prior to it should be reported. If the letters are all in non-increasing order, the program should be silent.

For example,

```
mac: ./letterOrder
v
q
q
n
q
# ‘q’ > ‘n’
p
R
P
^D
mac:
```

Note: The line starting with # is the only line generated by letterOrder. All the other lines are typed in by users. You may assume that all the input lines are valid.

```
/* Program to check whether the input letters are ordered*/
#include <stdio.h>
#include <stdlib.h>

int
main(int argc, char **argv) {
  int prev, cur, has_prev = 0;
  while((cur = getchar()) != EOF) {
    if (has_prev && cur > prev) {
      printf(“# ‘%c’ > ‘%c’\n”, cur, prev);
    }
    prev = cur;
    has_prev = 1;
    getchar(); /* To get rid of the <Enter> key */
  }
  return 0;
}
```

3. **[3 marks]** Write a *recursive* function double sum_sequence(int n) to calculate and return the sum:

$$\frac{1}{1}+\frac{1+2}{1*2}+\frac{1+2+3}{1*2*3}+\cdots+\frac{1+2+\cdots+n}{1*2*...*n}$$

You may assume that n is a positive number. For example, the call sum_sequence(3) should return the value (1/1) + (3/2) + (6/6) = 3.5.

```
double
sum_sequence(int n) {
  if (n == 1) {
    return 1;
  }
  int i;
  double num = 0, den = 1.0;
  for (i = 1; i <= n; i++) {
    num += i;
    den *= i;
  }
  return sum_sequence (n-1) + num/den;
}
```

End of sample exam