THE UNIVERSITY OF MELBOURNE
SCHOOL OF COMPUTING AND INFORMATION SYSTEMS

MID-SEMESTER TEST SAMPLE ANSWER
COMP10002 FOUNDATIONS OF ALGORITHMS

Total marks for this Exam: 10

Reading Time: **5** minutes
Writing Time: **40** 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]** Consider the following program execution:

```
mac:./starTriangle
Enter an integer: 5
*
**
***
****
*****
mac:
```

This program reads an integer n, and then prints a triangle with n rows of '*' characters. Complete the following program to implement the above process. You can declare more variables if necessary.

```
/* Program to print a '*' triangle */
#include <stdio.h>
#include <stdlib.h>

int
main(int argc, char **argv) {
        int n;
        int i, j;

        printf("Enter an integer: ");

        if (scanf("%d", &n) != 1) {
                exit(EXIT_FAILURE);
        }

        for (i = 0; i < n; i++) {
                for (j = 0; j <= i; j++) {
                        printf("*");
                }
                printf("\n");
        }

        return 0;
}
```

2. **[4 marks]** Write a function int count_factors(int n) that calculates and returns the number of factors, including 1 and itself, of the argument variable n.

   For example, the call count_factors(15) should return 4 (factors of 15 being 1, 3, 5, and 15). You may assume n > 0.

```
int
count_factors(int n) {
    int i, counter;
    counter = 0;
    for (i = 1; i <= n; i++) {
        if (n % i == 0) {
            counter++;
        }
    }
    return counter;
}
```

3. **[3 marks]** The following functions describe the computation behavior of a number of algorithms:

   (a) $f_1(n) = 3n - 2$
   (b) $f_2(n) = n - 4\log n + 5$
   (c) $f_3(n) = f_1(n) + f_2(n)$
   (d) $f_4(n) = f_1(n) - f_2(n)$
   (e) $f_5(n) = f_1(n) \times f_2(n)$
   (f) $f_6(n) = f_1(n) / f_2(n)$

   Using the "big-Oh" notation, what is the *best description* of the asymptotic growth rate of each of the functions $f_1()$ to $f_6()$?

(a) $f_1(n) = 3n - 2 \in O(n)$
(b) $f_2(n) = n - 4\log n + 5 \in O(n)$
(c) $f_3(n) = 4n - 4\log n + 3 \in O(n)$
(d) $f_4(n) = 2n + 4\log n - 7 \in O(n)$
(e) $f_5(n) = 3n^2 +$ *smaller terms* $\in O(n^2)$
(f) $f_6(n) = 3 +$ *smaller terms* $\in O(1)$ Note that you don't have to work out the details of the division at all. This is another reason why we like asymptotic analysis rather than doing exact arithmetic.

End of sample exam