CIS 22B
Class Assignment
Review - ANSWERS

A. Naming conventions: look at the following C++ identifiers and decide what should they be used for: a function name, a variable name, a file name, a named constant, etc.:

`length` – variable

A variable name consists of mainly lowercase letters; it must begin with a lowercase letter or '_')

Other examples: `stu_id`, `stuId`

`PI` – named constant (all uppercase letters)

Another example: `MAX_STU`

`sortScores` – function

A function represents an action: its name should include a verb. The only difference between a function name and a variable name is the "verb" included in a function name.

Another example: `binary_search`

B. What random numbers are generated by the following expression?

```
10 + rand() % 3
```

`rand()` returns a random integer from 0 to the largest possible integer
`rand() % 3` can be either 0, 1, or 2. These are all possible remainders when we divide by 3.
`10 + rand() % 3` evaluates to 10, 11, or 12.

In other words, integers within the range 10 to 12 inclusive.

C. What random numbers are generated by the following expression?

```
10 + 5 *(rand() % 3)
```

If the remainder is 0, we get: `10 + 5 * 0 => 10`
If the remainder is 1, we get: `10 + 5 * 1 => 15`
If the remainder is 2, we get: `10 + 5 * 0 => 20`

In other words, integers within the set {10, 15, 20}.

D. What random numbers are generated by the following expression?

```
rand() % (MAX - MIN + 1) + MIN
```

How many integer numbers are within the range [MIN, MAX]?
There are MAX – MIN + 1 integers within this range. All possible remainders when we divide by (MAX – MIN + 1) are: 0, 1, 2, … MAX – MIN. In other words, integers within the range 0 to MAX – MIN inclusive. By adding MIN we are shifting the range to MIN to MAX inclusive:

0 + MIN => MIN
MAX – MIN + MIN => MAX

Example: If MIN = -5, and MAX = 10, this formula will generate an integer within the range -5 to 10 inclusive: `rand() % 16 - 5`

E. Challenging problem: A file contains **N** distinct numbers within the range **0** to **N** inclusive. Find the missing number.

a. Example: Assume the file has the following numbers: **3, 0, 1, 5, 4.** For this file, N is 5 and the missing number is **2**.

b. Requirement: Design an algorithm to solve this problem in an effective way. Describe the algorithm in your own words (like a "to do list"). Writing code is OK but it is not required.

1st Solution

- Read data from file into an array: **3, 0, 1, 5, 2**
- N is the amount of numbers in the file, 5 in this example.
- Sort the array in ascending/descending order (it doesn't matter): **0, 1, 2, 3, 5**
- Traverse the array and compare consecutive elements: their difference should be 1, except for the missing number:
  1 – 0 = 1
  2 – 1 = 1
  3 – 2 = 1
  5 – 3 = 2: the missing number is either 3 + 1 or 5 – 1.

2nd Solution

- While reading numbers from file calculate their sum: **3 + 0 + 1 + 5 + 2 = 11**
- N is the amount of numbers in the file, 5 in this example.
- The sum of the first N natural numbers can be evaluated using the formula:
  N * (N + 1) / 2. In this case sumAll = 5 * (5 + 1) / 2 = 5 * 6 / 2 = 15
- Calculate the missing number: sumAll – sum // 15 – 11 => 4!

Which solution is better? The second solution is the best! It is elegant and efficient. It solves the problem with only one loop, without using a lot of memory. It works well even with a large file that is too big to be stored in an array. Math could be helpful sometimes in creating better algorithms!