TIC1001—Introduction to Computing and Programming I
National University of Singapore
School of Continuing and Lifelong Education

Practical Examination 1

21 September 2019

Time allowed: 2 hours

Instructions (please read carefully):

1. This is **an open-book exam**. You are allowed to bring in any course or reference materials in printed form. No electronic media or storage devices are allowed.
2. This practical exam consists of **two** questions. The time allowed for solving this test is **2 hours**.
3. The maximum score of this test is **20 marks**. Note that the number of marks awarded for each question **IS NOT** correlated with the difficulty of the question.
4. You are advised to attempt all questions. Even if you cannot solve a question correctly, you are likely to get some partial credit for a credible attempt.
5. While you are also provided with the templates `pe1-template.cpp`to work with, your answers should be submitted on Coursemology. Note that you can **only run the test cases on Coursemology for a limited number of tries** because they are only for checking that your code is submitted correctly. You are expected to test your own code for correctness locally on VS Code and not depend only on the provided test cases. Do ensure that you pasted your answers correctly by running the test cases at least once.
6. In case there are problems with Coursemology.org and we are not able to upload the answers to Coursemology.org, you will be required to name your file `pe1-<student no>.c` where `<student no>` is your student number and leave the file on the Desktop. If your file is not named correctly, we will choose any .cpp file found at random.
7. Please note that it shall be your responsibility to ensure that your solution is submitted correctly to Coursemology and correctly left in the desktop folder at the end of the examination. Failure to do so will render you liable to receiving a grade of **ZERO** for the Practical Exam, or the parts that are not uploaded correctly.
8. Please note that while sample executions are given, it is **not sufficient to write programs that simply satisfy the given examples**. Your programs will be tested on other inputs and they should exhibit the required behaviours as specified by the problems to get full credit. There is no need for you to submit test cases.
9. Marks will also be awarded for **good coding style** like suitable variable names, proper indentations, etc., in addition to the correctness of the program.

ALL THE BEST!

Question 1: Nine-pin Bowling [20 marks]

Nine-pins bowling is a game where a player rolls a ball to strike at nine bowling pins. In our version of nine-pin bowling, players only have one chance to knock down nine pins before they are reset after every bowl. (This is different from the commonly played ten-pin bowling where players get two tries to knock down ten pins.)

A game of nine-pins can thus be represented as an integer, with each digit indicating the number of pins knocked down in a bowl. The most significant digit (leftmost) represents the first bowl and the least significant digit (rightmost) represents the last bowl.

For example, a player with a game of 5 bowls where he knocks down 4 pins in his first bowl, and one more pin every subsequent bowl would be represented by the number `45678`. A player scoring a perfect game of 10 bowls would be `9999999999`.

Note: Because C/C++ interprets numbers with a leading 0 to be in octal (base-8), leading zeros should be omitted from the number. Thus, a gane where the player bowled 9 pins on the last bowl, but zero in all earlier bowls would simply be represented as the number `9`.

A. [Warm up] Without using any additional math libraries, implement the function

```
int pins_felled(long long game)
```

that takes as input a game and returns the total number of pins knocked down in the game. [6 marks]

Note the use of `long long` type in order to accept a game of ten bowls. You may assume the game will not exceed the allowable range of the type.

Sample tests:

Test function	Output
`pins_felled(45678)`	`30`
`pins_felled(9999999999)`	`90`

B. To reward players who manage to hit all nine pins in a bowl, the number of pins felled in the next bowl will be added to the score.

For example, the game `4927` will be $4+9+2+2+7=24$ as the score for the second bowl is $9+2$.

The score for a perfect game of `9999999999` will be 171, because each bowl except for the last bowl is worth $9+9=18$ points. $(9+9)\times 9+9=171$.

The score for the game `49090` is $4+9+9=22$ because after knocking down nine pins in a bowl, the player did not knock down any in the next bowl.

Without using any additional math libraries, implement the function

```
int score(long long game)
```

which takes in a game and returns the score of the game using this scoring scheme. [7 marks]

Sample tests:

Test function	Output
`score(45678)`	`30`
`score(4927)`	`24`
`score(9999999999)`	`171`

C. Another scoring scheme awards bonus score to consecutively felling the same numer of pins. The bonus score awarded is the number of pins felled, multiplied by the number of consecutive previous bowls of the same pins felled.

For example, suppose the player fells 9 pins over 3 consecutive bowls. For the first bowl, the score will remain as 9. For the second bowl, the score will be $2 \times 9 = 18$ as it is the second consecutive 9 pins felled. For the thrid bowl, the score will be $3 \times 9 = 27$ as it is the third consecutive 9 pins felled. So the total score for the three bowls is $9 + 18 + 27 = 45$.

Another example, the following game will have a total score of 81:

$$\begin{array}{rl} \text{pins felled:} & 3 \quad 3 \quad 5 \quad 5 \quad 8 \quad 8 \quad 8 \quad 3 \quad 3 \\ \text{score per bowl:} & 3 + (2\times 3) + 5 + (2\times 10) + 8 + (2\times 8) + (3\times 8) + 3 + (2\times 3) = 81 \end{array}$$

Without using any additional math libraries, implement the function

```
int consec_score(long long game)
```

which takes in a game and returns the score of the game using this consecutive scoring scheme.

[7 marks]

Sample tests:

Test function	Output
`consec_score(335588833)`	`81`
`consec_score(9999999999)`	`495`

— E N D O F P A P E R —