CS3353: Data Structures and Algorithm Analysis I
Spring 2024

Homework #3

- Full name only: _____
- Release date: 4:00 PM, Feb 21, 2024 (Wednesday)
- Due date: **2:30 PM, March 04, 2024 (Monday)**
- It should be done INDIVIDUALLY; Show ALL your work; Submit your source code and results through Canvas.
- Please refer to the course syllabus (Course Requirements and Grading Policy - Assignment, page 3) for the policy of submission, late submission, missing submission, and wrong submission.
- Total: 20 pts
- **Grading Policy**
    - Stack implementation in linked list (6 pts)
        - Stack implementation correctly: 6 pts
        - Stack implemented incorrectly: -3 pts; TA needs to evaluate the program.
            - Stack class: 1 pt
            - Pop: 1 pt
            - Push: 1 pt
                - Partial points (e.g., 0.5 pt) are deducted depending on the degrees of wrongness.
        - Stack implementation in non-linked list: -3 pts
        - No stack implementation: -5 pts
    - Binary number conversion (4 pts)
        - Correct output: 4 pts
            - Using Stack: 4 pts
            - Not using Stack: -3 pts
        - Incorrect output:
            - Simple mathematical calculation/formula error: -1
            - The idea/logic of algorithm is wrong: -3
                - Partial points (e.g., 2 pts) are deducted depending on the degrees of wrongness.
    - Octal number conversion (4 pts)
        - Correct output: 4 pts
            - Using Stack: 4 pts
            - Not using Stack: -3 pts
        - Incorrect output:
            - Simple mathematical calculation/formula error: -1
            - The idea/logic of algorithm is wrong: -3
                - Partial points (e.g., 2 pts) are deducted depending on the degrees of wrongness.
    - Hexadecimal number conversion (4 pts)
        - Correct output: 4 pts
            - Using Stack: 4 pts
            - Not using Stack: -3 pts
        - Incorrect output:
            - Simple mathematical calculation/formula error: -1
            - The idea/logic of algorithm is wrong: -3
                - Partial points (e.g., 2 pts) are deducted depending on the degrees of wrongness.
    - Self-testing results (2 pts)
        - No testing conducted / No WORD document: -2
    - Compilation error / run-time error:
        - Deduct 5 pts first. Then TA needs to evaluate the program and apply the above grading policy.
    - No homework submission: 0 pts

- o   Students will not receive 0 pts if students spent time and effort on program and make the submission.
    - If the program has issues/problems, TA needs to evaluate student's program and gives partial points depending on the quality/completion of program.
  - o   The instructor will decide the grade policy of any scenario which is not covered by the above list. Meanwhile, please kindly contact the instructor if you have any questions regarding the grading policy.

1. Write a program to convert a number from a decimal notation to a number expressed by a number system whose base is 2 (binary), 8 (octal), or 16 (hexadecimal). The conversion is performed by repetitious division by the base to which a number is being converted and then taking the remainders of division in the reverse order. For example, in converting to binary, number 6 requires three such divisions: 6/2 = 3 remainder 0, 3/2 = 1 remainder 1, and finally, 1/2 = 0 remainder 1. The remainders 0, 1, and 1 are put in a reverse order so that the binary equivalent of 6 is equal to 110. Here is a set of requirements to follow:

- • Type the homework number and your full name at the top in your all source codes.

```
/* Homework #3, James Bond */
```

- • A hexadecimal system requires 16 digits: 0, 1, …, 9, A, B, C, D, E, F. In this system, decimal number 26 is equal to 1A in hexadecimal notation because 26/16 = 1 remainder 10 (that is, A), and 1/16 = 0 remainder 1.
- • Your program should be **menu-driven** and **execute the chosen command**. If you type 3, then exit the program.

```
                    M E N U

        Binary (0), Octal (1), Hexadecimal (2)
        Exit Program (3)

                    Choose?
```

- • Deploy a stack that is implemented by a linked list.
- • For example,

```
                M E N U

    Binary (0), Octal (1), Hexadecimal (2)
    Exit Program (3)

                Choose?   0   6

1 1 0

                M E N U

    Binary (0), Octal (1), Hexadecimal (2)
    Exit Program (3)

                Choose?   1   15

1 7

                M E N U

    Binary (0), Octal (1), Hexadecimal (2)
```

```
                    Exit Program (3)

                          Choose?  2  26

        1 A
                .
                .
                .
```

- Submit your source code and self-testing results (e.g., readable and clear screenshots) through Canvas before the due date, **2:30 PM, March 04, 2024 (Monday)**. The TA will build and run your source code and test with random input.
  - **Source code (one file only)** – The file name should be "your name + homework number", e.g., james_bond_3.cpp or james_bond_3.java.
  - **Self-testing Results** (e.g., readable and clear screenshots) **in WORD** document