McMaster University
MECHTRON 2MD3: Data Structures and Algorithms for Mechatronics

Winter 2023
Midterm take home test
Due: March 4, 2023 at 12:30 pm

Instructions
All questions annotated “short answer” should be typed and submitted in a single pdf document with the file name “1234567-midterm.pdf” where 1234567 is your student ID. Each question annotated “programming” should be submitted as a single C++ source file named “1234567-midterm-x.cpp” where 1234567 is your student ID and x is the question number. All code in programs and short answers should be formatted and commented. Please ensure your source files compile and run properly before submitting.

Questions

1. [programming, 10 marks] Reverse Polish Notation (also known as postfix notation) is an unambiguous way of writing an arithmetic expression without parentheses. It is defined so that if "(exp_1) ∘ (exp_2)" is a normal fully parenthesized expression whose operation is "∘", then the postfix version of this is "$pexp_1$ $pexp_2$∘", where $pexp_1$ is the postfix version of exp_1 and $pexp_2$ is the postfix version of exp_2. The postfix version of a single number or variable is just that number or variable. So, for example, the postfix version of "((5 + 2) * (8 − 3))/4" is "5 2 + 8 3 − * 4 /". Write a program that uses a stack for evaluating an expression in postfix notation. Wherever possible, you should use the object-oriented design patterns discussed in class. You must implement:
    a. A class named Stack that implements the stack ADT.
    b. An RPNEvaluator class that uses your Stack class.
    c. Your program must use this main method:

```
int main() {
  std::string rpn_exp;
  std::cout << "Please enter an expression in RPN format: ";
  std::getline(std::cin, rpn_exp);
  RPNEvaluator rpn(rpn_exp);
  rpn.Evaluate();
  return EXIT_SUCCESS;
}
```

Example input: 5 2 + 8 3 − * 4 /
Expected output: Answer: 8.75

Each part of the input expression is separated by one or more spaces, and you can expect input to contain only integers and operators.

You must handle the following operators: + - * /
If the input expression is malformed or contains input other than integers and operators, your program should output this string "Error: malformed expression".

2. [short answer, 5 marks] Al and Bill are arguing about the performance of their sorting algorithms. Al claims that his O(n log n)-time algorithm is always faster than Bill's $O(n^2)$-time algorithm. To settle the issue, they implement and run the two algorithms on many randomly generated data sets. To Al's dismay, they find that if $n < 100$ the $O(n^2)$-time algorithm actually runs faster, and only when $n \geq 100$ the O(n log n)-time one is better. Explain why the above scenario is possible. You may give numerical examples.

3. [programming, 10 marks] Implement a capacity-limited version of the deque ADT based on a vector used in a circular fashion. Your class should be called Deque and support the following methods. It will store std::string elements.
    a. Deque(int *n*):
       Construct a deque with capacity *n*.
    b. insertFront(std::string *e*):
       Inserts a new element *e* at the beginning of the deque.
    c. insertBack(std::string *e*):
       Inserts a new element *e* at the end of the deque.
    d. eraseFront():
       Removes the first element of the deque (if it is not empty)
    e. eraseBack():
       Removes the last element of the deque (if it is not empty)
    f. front():
       Returns the first element of the deque. If the deque is empty, the string "error: empty deque" is returned.
    g. back():
       Returns the last element of the deque; If the deque is empty, the string "error: empty deque" is returned.
    h. size():
       Return the number of elements in the deque.
    i. empty():
       Return true if the deque is empty and false otherwise.

    Your class must be called "Deque" and it must include the above methods with those exact names. You will need to design your own main method to test your Deque, but your submission must NOT include a main method.

4. [short answer, 4 marks] Write a single tail recursive C++ function that will rearrange a vector of int values so that all the even values appear before all the odd values.

5. [short answer, 2 marks] Show by counterexample that the following statement is false: For any positive constant c, f(cn) = O(f(n)).

6. [short answer, 6 marks] Part 1: Suppose it is known that the running time of an algorithm is $(1/3)n^2 + 6n$, and that the running time of another algorithm for solving the same problem is $111n - 312$. Which one would you prefer, assuming all other factors equal? Part 2: Of course, "all other factors" are never exactly equal for two algorithms. There is no question that execution time is important, but there are other factors to consider when choosing an algorithm. Can you think of other considerations ? (Provide 3 other considerations for full marks)

The End.