**CS 336: Algorithms** **Problem Set 1**

**Date:** Thursday, September 23, 2021 **Due:** Thursday, September 30, 2021

**Submit your solution on Canvas.**

**Problem 1** (Quiz.)**.** Answer the following questions. Enter your answers on Canvas under *Quizzes*. We will open the Canvas submission page on September 27.

1. True or False: $n = O(n \log_2 n)$.
2. True or False: $n^2 = O(n)$.
3. True or False: $n \log_2^5 n = O(n^2)$.
4. True or False: $\log_2 n = O(\log_e n)$.
5. True or False: $\log_e n = O(\sqrt{\log_2 n})$.
6. True or False: $2^n = O(n^{\log_2 n})$.
7. True or False: $n^{\log_2 n} = O(2^n)$.
8. True or False: For all positive functions $f$ and $g$, if $f(n) = O(g(n))$, then $n \cdot f(n) = O(n \cdot g(n))$.
9. True or False: For all positive functions $f$ and $g$, if $f(n) = O(g(n))$, then $f(n) + n = O(g(n) + n)$.

**Collaboration policy for Problem 1:** Please, solve this problem on your own. Do not collaborate with other students.

**Problem 2.** Alice and Bob are planning to go to a restaurant to celebrate Alice's birthday. Alice chose a place from a list of restaurants (`restaurantList`), encrypted the name of this place using a *ToyCrypt* cipher, and then sent the encrypted name to Bob. Bob received the encrypted message. However, he does not know the encryption key, so he cannot decrypt the message. Your goal is to write a computer program that finds the key.

The ToyCrypt cipher uses keys that are integer numbers from 0 to $n$, where $n$ is a parameter which is less than 100000. Hence, you can find the key using the *exhaustive search*. In other words, you can try all possible keys from 0 to $n$ and return the one that decrypts the message. To decrypt the message use function

```
std::string decrypt(const std::vector<int>& encryptedName, int key),
```

which decrypts message `encryptedName` using key `key`. You can assume that the message is decrypted correctly if the resulting string is in the list of restaurants `restaurantList`.

Please, implement the following function

```
FindKey(const std::vector<int>& encryptedName, int n, const std::string& restaurantList)
```

Function `FindKey` should return the key that was used by Alice to encode the restaurant name. The arguments of the function are as follows:

- `const std::vector<int>& encryptedName` – the encrypted message;

- `int n` – the maximum possible value of the key (i.e., the key must be an integer in the range from 0 to $n$);
- `const std::string& restaurantList` – the list of restaurants separated by “;”.

Your program should be fast! It should pass all tests in no more than 5 seconds (in our test environment).

**Collaboration policy for Problem 2:** You can discuss this problem with other students. You can also ask other students questions about C++. However, you need to write your code completely on your own.

**Instructions for the programming assignment.**

Download files

- `student_code_1.h` – this file should contain your solution.
- `problem_solver_1.cpp` – this is the main file in the project (don’t edit this file!).
- `test_framework.h` – this is a library responsible for reading and writing data files (don’t edit this file!)
- `small_problem_set_1.in` and `large_problem_set_1.in` – these files contain test problems for your algorithm (don’t edit these files!)

Place all files in a new folder/directory. Write your code in function `FindKey`. Also, write your name in the function `GetStudentName`. Both functions are located in file `student_code_1.h`. Compile and run your code. To compile your code do the following.

- If you use Clang compiler, type
  ```
  clang++ -std=c++17 -pedantic-errors problem_solver_1.cpp -O2 -o problem_solver_1
  ```
- If you use GNU C++ compiler, type
  ```
  g++ -std=c++17 -pedantic-errors problem_solver_1.cpp -O2 -o problem_solver_1
  ```
- If you use Microsoft Visual C++ compiler, start `Developer Command Prompt` and type
  ```
  cl /EHsc problem_solver_1.cpp
  ```

Your compiler should be compatible with `C++17`. If you work in the Wilkinson Lab, you need to start developer tools first: Type

- `scl enable devtoolset-4 bash`

Once you compile your code, start your program. Type `./problem_solver_1 small` to run your code on simple problems and `./problem_solver_1 large` to run your code on hard problems. On Windows, type `problem_solver_1.exe small` and `problem_solver_1.exe large`, respectively. Make sure that executable is located in the same folder as files `small_problem_set_1.in` and `large_problem_set_1.in`. If your code works correctly, you will get the following message:

> Problem set 1. Your algorithm solved all test problems correctly. Congratulations! `solution_1.dat` via Canvas.

If your code makes a mistake, you may get a message like this:

```
Problem set 1.  Mistake in problem #15.  Correct answer:  4.  Your answer:  12.
```

Please, test your code with the both problem sets (small and large). When your code is ready, submit file `student_code_1.h` on Canvas. Make sure that you are submitting the latest versions!