Algorithms EECS 336 **Problem Set #3**
Date: April 22, 2018 **Due:** April 29, 2018

Submit your solution on Canvas.
Do not discuss these problems with other students. You should solve these problems on your own.

Problem 1. You are asked to design a computer program for a student film club. Every week, the board of the club picks a new movie and shows it to the club's members. To accommodate busy schedules of its members, the club has several screenings per week. The screenings are chosen in such a way that each member of the club can attend at least one of them.

In the beginning of the week, each person specifies a time frame that works for that person (e.g., Alice may submit an interval from 2:10pm till 6pm on Monday, while Bob can submit an interval from 8pm till 11pm on Friday). Your program should find the minimum number of screenings such that each club member could watch the movie in the interval of time he or she specified.

Formally, the algorithm receives a set of intervals $[l_i, r_i]$ and a parameter T. The interval $[l_i, r_i]$ is the time that works for person $\#i$. Parameter T is the length of the movie. The algorithm needs to find the minimum set of intervals $[s_j, f_j]$ of length T each such that for each i, there exists interval $[s_j, f_j]$ such that

$$[s_j, f_j] \subset [l_i, r_i].$$

Each interval $[s_j, f_j]$ corresponds to the time of the j-th screening of the movie. You may assume that the time is measured in minutes after midnight on Sunday and all numbers a_i, b_i, s_j, f_j are integers.

I. Design and describe a greedy algorithm for this problem.

II. Analyze its running time. To get a full credit for the problem, the running time of the algorithm must be $O(n \log n)$.
III. Prove that the algorithm is correct.

Problem 2. Implement your algorithm from the previous problem. You can use one of the following function signatures:

- `int FindBestSchedule (std::vector<std::pair<int,int>> availabilities, int T)`
- `int FindBestSchedule (const std::vector<int>& left, const std::vector<int>& right, int T)`

In the first case, the function `FindBestSchedule` receives a list of pairs of integers (`availabilities`). The first value in the pair $\#i$ (`availabilities[i].first`) is the left end of the interval $\#i$; the second value (`availabilities[i].second`) is the right end of the interval $\#i$. Parameter T is the length of the movie. In the second case, the function `FindBestSchedule` receives two arrays of integers: `left` and `right`. The left endpoint of the interval $\#i$ is `left[i]` and the right endpoint is `right[i]`. In both cases, the function should return the number of screenings in the optimal solution.

Instructions for the programming assignment. Download files:

- `student_code_3.h` – this file should contain your solution.
- `problem_solver_3.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!)

- `problem_set_3.dt` – this file contains test problems for your algorithm (don't edit this file!)

Place all files in a new folder/directory. Write your code in the function `FindBestSchedule`. Also, write your name in the function `GetStudentName`. Both functions are located in file `student_code_3.h`. Compile and run your code. To compile your code do the following.

- If you use GNU C++ compiler, type
  `g++ -std=c++11 problem_solver_3.cpp -o problem_solver_3`
- If you use CLang compiler, type
  `clang++ -std=c++11 problem_solver_3.cpp -o problem_solver_3`
- If you use Microsoft Visual C++ compiler, start `Developer Command Prompt` and type
  `cl /EHsc problem_solver_3.cpp`

Your compiler should be compatible with `C++11`. If you work in TLab, you need to start developer tools first: Type

- `scl enable devtoolset-4 bash`

Once you compile your code, start your program. Type `./problem_solver_3` on Unix or Mac and `problem_solver_3.exe` on Windows. Make sure that the executable is located in the same folder as file `problem_set_3.dt`. Your program will generate `solution_3.dat` that contains solutions to the problems from file `problem_set_3.dt`. If your code works correctly, you will get the following message:

- Problem set 3. Your algorithm solved all test problems correctly. Congratulations!
- Don't forget to submit your source code and file `solution_3.dat` via Canvas.

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

- `Problem set 3. Mistake in problem #15. Correct answer: 4. Your answer: 12.`

Finally, when your code is ready, submit files `student_code_3.h` and `solution_3.dat` via Canvas. Make sure that you are submitting the latest versions.

Remark: If you want to debug your code, please, type `./problem_solver_3 15` on Unix or Mac and `problem_solver_3.exe 15` on Windows. This command will call your function only on one problem – the problem #15 and thus let you debug your code on the problem where your program erred. Note that this command will not generate or update `solution_3.dat`. So before submitting your solution, you need to run your program without any command line arguments.