

# Programming Design, Spring 2013

## Homework 07

Instructor: Ling-Chieh Kung  
Department of Information Management  
National Taiwan University

To submit your work, please upload the following one file to the online grading system PDOGS at <http://stella.im.ntu.edu.tw/online-judgement/>.

1. A PDF file for Problems 1, 2, and 3.

NO hard copy and NO late submission. The due time of this homework is 1:00pm, April 15, 2013.

### Problem 1

(10 points; 5 points each) In the C++ standard library `<ctime>`, a function `clock()` is defined. This function returns the number of “di-da” (it is just a time unit shorter than a second), as a `long int`, that have past since the execution of the program. One second consists of `CLK_TCK` di-da, where the constant `CLK_TCK` is also defined in `<ctime>`. Below we will demonstrate how to utilize `<ctime>` and `CLK_TCK` to calculate the amount of time one needs to execute a set of statements.

Consider the following program:

```
long int stTime = 0, endTime = 0;
double duration = 0;
stTime = clock();

// statements to run start here
// ...
// statements to run end here

end_time = clock();
duration = static_cast<double> (endTime - stTime);
duration = duration / CLK_TCK;
cout << duration << endl;
```

- (a) What will this program print out?
- (b) Why do we need to cast `endTime - stTime`?

### Problem 2

(40 points) Consider the program “PDSp13\_hw07\_pointer.cpp”:

- (a) (5 points) Suppose an array {7, 4, 5, 8, 2, 6, 3} is randomly generated and input into the function `maxPtr()`. What will be the output of this program?
- (b) (5 points) What does the function `maxPtr()` do in general? Answer this question by explaining how the function process the input parameters.
- (c) (10 points) What does the statement `cout << ptrGMax - value << endl;` do in general? To answer this question, you need to explain the physical meaning of `ptrGMax - value`.
- (d) (10 points) The function `maxPtr()` is correct, but it is not implemented in the most efficient way. Propose a way to improve the efficiency. If you believe your description is clear and precise enough, you do not need to write a program to demonstrate your idea.

- (e) (10 points) Write a function that can do exactly the same thing as `maxPtr()` does without using pointers. After the execution of this new function, the main function should still be able to take the outputs of the new function to print out the two pieces of information originally printed out by `cout << *ptrGMax << endl;` and `cout << ptrGMax - value << endl;`. Do not submit a .cpp file for this problem. Just include the codes for the function into your .pdf file. You do not need to explain how to invoke this function.

**Hint.** You may want to return multiple values from your new function.

### Problem 3

(50 points) Consider the program “PDSP13\_hw07\_rand.cpp”. In this program, we implement two algorithms for generating `ARRAY_LEN` nonrepeating random numbers from 0 to `ARRAY_LEN - 1`.

- (a) (0 points) Convince yourself that both algorithm run correctly. You may want to set `ARRAY_LEN` to be small and invoke the function `print()`.
- (b) (5 points) Try to set the initial value of `ARRAY_LEN` to be 2500, 5000, 7500, 10000, ..., and 30000. For each of these values, compare the execution time of algorithms 1 and 2.
- (c) (15 points) Explain in words how these two algorithm work. Your explanations need not to be long. All you need is to precisely describe the ideas of these two algorithms.
- (d) (10 points) Which algorithm do you think is better? Why? If one is more efficient than the other, what is the main difference?
- (e) (10 points) Rewrite the function `bruteForce()` by replacing the outer loop by a `while` loop. Do not submit a .cpp file for this problem. Just include the codes for the function into your .pdf file.
- (f) (10 points) Try to set the initial value of `ARRAY_LEN` to 32500. If the program terminates successfully, set it to 35000. What happens when you execute the program? Why does that happen?