

COMP2113 Programming technologies [2019]

Assignment 3

General Instructions

Please read the instructions in this document carefully. In this assignment, you will solve **ONLY ONE** tasks and a tester would automatically test your submitted program. Sample test cases are provided with this task in this document (**user inputs are printed in blue**). However, please note that we will also use additional test cases when marking your assignment submission.

Total: 100 points

5 points for proper code comments and indentation

5 points for program modularity using appropriate functions

90 points for program correctness

Deadline

The deadline is announced in Moodle. **Late submission will not be accepted.**

Late submission

You will receive 0 marks if you submit after the deadline.

Wrong submission

You will receive 0 marks if you submit incorrect files.

Evaluation

Your code will be tested for technical correctness. However, the correctness of your implementation – not the tester's judgments – will be the final judge of your score. If necessary, we will review your work individually to ensure that you receive due credit for your work.

Academic Dishonesty

We will be checking your code against other submissions in the class and from the Internet for logical redundancy. If you copy someone else's code and submit it with minor changes, we will know. We trust you all to submit your own work only; please don't let us down. If you do, we will pursue the strongest consequences available to us.

Task 1 (and the only one task) Game of Elimination

(100 points)

Consider a prize to be awarded to a winner among $n > 0$ contestants by a game of elimination. The n contestants first line up in a line, and are assigned the numbers 1 to n one by one.

The host of the game then decides on an integer $k > 1$, and starting from contestant 1, the host counts k contestants down the line and eliminates the $(k+1)$ -th contestants from the circle. He then continues to count for another k contestants, and eliminates the $(k+1)$ -th contestants from the line. When he counts to the end of line, he will continue counting from the beginning of the line. This process repeats until there is only one contestant remains who will be the winner of the prize.

For example, if $n = 7$ and $k = 3$,

The initial line: 1234567, count to 3 and contestant 4 is eliminated

Line now becomes: 123567, continue counting from 5 and contestant 1 is eliminated

Line now becomes: 23567, continue counting from 2 and contestant 6 is eliminated

Line now becomes 2357, continue counting from 7 and contestant 5 is eliminated

Line now becomes 237, continue counting from 7 and contestant 7 is eliminated

Line now becomes 23, continue counting from 2 and contestant 3 is eliminated

Winner is contestant 2

Write a C++ program which implements a circular linked list (see Module 8 Guidance Notes p.89) to determine which contestant will win the prize given two input numbers n and k ($n > 0$ and $k > 1$).

Important Note: You will need to implement the circular linked list on your own and you are NOT allowed to use any data structures from the STL libraries.

Idea:

- Use the program `build_list_forward.cpp` of Module 8 as a basis for your work. The program essentially has built a linked list; to turn it into a circular linked list, you just need to point the next pointer of the last node to the head of the list.
- You will need to build a circular linked list containing the numbers 1 to n
- You will then need to traverse through the linked list and remove a node once you have traversed for k nodes.
- After removing a node, you should check if there remains only one node in the list. If yes,

the winner is identified.

- Remember to free all memories used by the linked list before the program terminates.

Hint:

- Study carefully build_list_forward.cpp and build_list_sorted.cpp of Module 8. They should contain the main building blocks for you to finish this task.
- It is expected that you will only need to write no more than 20-30 lines of additional code, after reusing the appropriate codes in build_list_forward.cpp and build_list_sorted.cpp. Of course, it is OK if you have written more than that.

Sample test case 1.1 (user input in blue)

7 3
2

Sample test case 1.2 (user input in blue)

5 2
4

Sample test case 1.3 (user input in blue)

1 6
1

Sample test case 1.4 (user input in blue)

10 4
3

Sample test case 1.5 (user input in blue)

124 2
58

Sample test case 1.6 (user input in blue)

100000 4
40333

Sample test case 1.7 (user input in blue)

123456 7
30705

Sample test case 1.8 (user input in blue)

8888888 100
549337

Submission:

Name your program as 1.cpp and save it into a directory. After completing the assignment, compress this directory as a .zip file. Please use **your university number** to name the zip archive and check carefully that the correct file have been submitted (your zip file should contain the only one file 1.cpp). We suggest you download your submitted file, extract them, and check for

correctness. **You will receive 0 marks for this assignment if you submit incorrect files.**

Resubmission after the deadline is not allowed.

Coding and marking environment

Please note that you must make sure that your program can compile, execute and generate the required outputs on our standard environment, namely, the gcc C++11 environment we have on the CS Linux servers (academy*). While you may develop your work on your own environment, you should always try your program (compile & execute & check results) on our standard environment before submission.

The compiler command `g++ -pedantic-errors -std=c++11 1.cpp` will be use to compile your program in the above standard environment.

Input and output format

Please follow the instructions given by the question regarding input/output requirements for your program. If you failed to follow the instructions, the auto-grader may not able to give a score for your program. Additionally, you should strictly follow the sample output format (including space, line breaker, etc.), otherwise, your answer might be considered as wrong.