

CS 158/159 Homework 6

This assignment is worth 15 points and will be **due Monday April 9, 2018 at 11:00pm**. All assignment deadlines are firm and the ability to submit your assignment will be disabled after the deadline elapses. No late work will be accepted. You are encouraged to start this assignment as soon as possible so that you have ample time to seek assistance should you experience difficulties completing or submitting this assignment.

This programming assignment does not have a single solution, and the assignment you submit must be your own original work. **Collaboration with other students is not permitted on homework assignments.** Any submission may be processed with comparison software and the results will be used to detect unacceptable collaboration between individuals. If you need assistance, you should only consult course staff regarding your program.

Your program must adhere to the course programming standards (available in the course packet and in Blackboard). Please review this document before submitting your assignment, because failure to adhere to it will result in a reduction in points. Your program must include the designated program header (`~cs15x/student/hdrProg`) at the top of the program (which can be inserted in `vi` using the `hp` shortcut while in command mode). The header must include an appropriate description of your program and must contain your official Purdue career account e-mail address. Also note that course standards prohibit the use of advanced programming concepts not yet introduced in the course, unless otherwise specified.

Each of the example executions provided below represents a single execution of the program. Your program must accept input and produce output **exactly** as demonstrated in the example executions. Your program will be tested with the data seen in the examples below and an unknown number of additional tests making use of reasonable data. Do not include any example outputs with your submission.

A single program file (with the `.c` extension) must be submitted electronically via the `guru` server. An example submission was conducted during the first week in lab00. Any attempts to submit via another method will be denied consideration. You may make multiple submissions before the deadline, but only the last attempt is retained and graded. All previous submissions will be over-written and cannot be recovered. The submission script will reject the submission of any file that does not compile. A program must compile to be considered for partial credit. You should always check the confirmation e-mail you receive after a submission to verify that you have submitted the correct file, to the correct assignment, and to the correct lab section. If you have a concern regarding how to submit work, please visit course staff prior to the assignment deadline.

Problem: A type-1 match is an integer with two consecutive digits that are the same. A type-2 match is an integer where two consecutive digits that are the same can be created by removing another single digit (example: 41013, remove the zero digit and the resulting value is 4113). Given a range of integers count the number of type-1 and type-2 matches. A number that has been determined to be a type-1 match will not be tested to determine if it is also a type-2 match.

Example Execution #1:

Enter starting value of range -> 100
Enter ending value of range -> 200

For data in the range: [100, 200]
Number of type-1 matches: 20
Number of type-2 matches: 9

Example Execution #2:

Enter starting value of range -> 0
Enter ending value of range -> 99

For data in the range: [0, 99]
Number of type-1 matches: 9
Number of type-2 matches: 0

Example Execution #3:

Enter starting value of range -> 2000000
Enter ending value of range -> 2001000

For data in the range: [2000000, 2001000]
Number of type-1 matches: 1001
Number of type-2 matches: 0

Example Execution #4:

Enter starting value of range -> 123456789
Enter ending value of range -> 123476789

For data in the range: [123456789, 123476789]
Number of type-1 matches: 6149
Number of type-2 matches: 5139

Example Execution #5:

Enter starting value of range -> 100
Enter ending value of range -> 1726354

For data in the range: [100, 1726354]
Number of type-1 matches: 757130
Number of type-2 matches: 386617

Example Execution #6 (input validation expectations demonstrated):

Enter starting value of range -> -5

Error! Starting value must be non-negative!!

Enter starting value of range -> 582

Enter ending value of range -> 550

Error! Ending value must be >= 582

Enter ending value of range -> 582

For data in the range: [582, 582]

Number of type-1 matches: 0

Number of type-2 matches: 0

Example Execution #7 (more input validation expectations demonstrated):

Enter starting value of range -> 2345678

Enter ending value of range -> 8234678

Error! The maximum difference between start and end is 5000000

Enter starting value of range -> 1

Enter ending value of range -> 5000001

For data in the range: [1, 5000001]

Number of type-1 matches: 2276367

Number of type-2 matches: 1164816

Additional Notes:

- All test cases will only include values that will fit into an `int` type variable.
- Input validation requirements:
 - a) The ending value must be greater than or equal to the starting value (see example #6).
 - b) The range cannot exceed 5,000,000 (see example #7).
- Course standards prohibit the use of programming concepts not yet introduced in lecture. For this assignment you can only consider material in the first 6 chapters of the book, notes, and lectures. Use of advanced programming constructs beyond this material would result in a loss of points.