CSE 11 Fall 2020 PA1 - Covid Transmission

Due date: Wed, Oct 14 @ 11:59PM PDT (Thurs, Oct 15 @ 11:59PM PDT with slip day)

No submission will be accepted after Thurs, Oct 15 @ 11:59PM PDT

Total Points: 100

(Not: If you submit your assignment late, the autograder will automatically use your slip day if you have any remaining.)

Academic Integrity:

We take Academic Integrity (AI) very seriously at UCSD. Before beginning the assignment please fill out the AI form:

https://forms.gle/K2pksB6KeF667zFbA

You must fill this out in order to recieve points for any assignments.

Consent to Act as a Research Subject (1 point):

Consent form for analzying students and tutors session videos:

https://forms.gle/UZAi4a4KbBvixem88

You must fill this out in order to recieve points.

Please note that participation in this research is completely voluntary. Your decision to participate or not participate will have no effect on your activities or grades in this class. However, participation in this research study will help the faculty better understand what topics and questions frequently come up in tutoring sessions, which will then help to train tutors. You must fill in the form even if you choose to not participate.

Preclass survey (1 point):

This survey will help us understand the background and goals of the students in this course. You will receive credit for submitting it, but your specific answers will not affect your grade in the course in any way.

https://forms.gle/uC5u26wrrAnDarCc9

Week1 survey for PA1 (1 point):

We ask that you spend a few minutes reflecting on your experience in this course this week. Please give thoughtful and truthful answers, to the best of your ability. Your specific answers will not affect your grade in any way. You will receive credit as long as you complete it.

https://forms.gle/ZCaY8Z7S22zb55kg8

Style:

On this assignment, we will give you feedback on style but **not deduct points** for problems with style. For future assignments, we will be grading the following for style on all files you submit:

1. File header
2. Class header
3. Method header(s)
4. Inline comments
5. Proper indentation
6. Descriptive variable names
7. No magic numbers
8. Reasonably short methods (if you have implemented each method according to specification in this write-up, you're fine). This is not enforced as strictly.
9. Lines shorter than 80 characters
10. Javadoc conventions (@param, @return tags, /** comments */, etc.)

A full style guide, https://docs.google.com/document/d/1xqhafV9rKQXRO1G2hq9S_O5JqK-1xQw2Fpf9fPJO-sI, can be found here. If you need any clarifications, feel free to ask on Piazza.

Setup:

We will be using the command line in this PA and the PAs to follow. Be sure you have access to bash:

- Running bash on Windows: https://gitforwindows.org/
- If you have a Mac machine open up the Terminal App

Now that you have bash ready to go familiarize yourself with some commands (optional):

- Essential Commands: https://www.hongkiat.com/blog/web-designers-essential-command-lines/
- Unix reference sheet: https://files.fosswire.com/2007/08/fwunixref.pdf

You will need java installed on your machine in order to do this assignment:

- Java Environment Setup: https://docs.google.com/document/d/1GentxxfsUOk0g7YsqBV8-spUEANboMFa9UWENe_yIRU/edit#heading=h.60uzkqq015s9

Files to Submit:

`CovidTransmission.java`

In order to receive points for this assignment, your file name **must** be `CovidTransmission.java`

Goal:

Programming Assignment 1 is an introduction to Java programming. In this PA, you will get exposure to primitive data types, variables, keyboard input, console output, and if-else logic.

As you get started, please pay attention to the following:

- Please read the **entire write-up** before getting started.

Logistics:

In EACH AND EVERY FILE that you turn in, we need the following in comments at the top of each file. These are essential so that we can more easily process your submissions and ensure that you receive proper credit. This is a very large class with hundreds of students when combining all lectures. **No name, no points**.

NAME: <your name>

ID: <your student ID>

EMAIL: <your email>

Problem Description: Covid Transmission (97 Points)

Disclaimer: These are sample statistics that do not accurately reflect covid transmission. Please follow the CDC guidelines to stay up to date on the coronavirus pandemic: https://www.cdc.gov/

The Centers for Disease Information (CDI) has recently found a way to compute risk for contracting COVID from someone who tested COVID positive. The risk is computed using the amount of time the two individuals spent together. However all the programmers that worked for the CDI left to work at Apple for much more money (and free snacks), so the CDI now needs your help to compute covid risks.

Using the amount of time that two individuals: person A and person B spend together the CDI can now tell people how likely person A is to have gotten covid from the infected person B. And they need you to convert their ideas into code.

Write a program called `CovidTransmission` that outputs the number of minutes that person A and person B are in contact with each other and the risk level of the virus transmitting.

Formula to Determine Risk Level:

0 mins <= contact time <= 60 mins: `low`

60 mins < contact time <= 180 mins: `medium`

180 mins < contact time <= 360 mins: `high`

contact time > 360 mins: `extremely high`

Edge Cases: **contact time < 0: set both contact time and risk level to -1**, i.e. output `-1 -1`.

Program Name: CovidTransmission

The file name should be `CovidTransmission.java`, and the class name should be `CovidTransmission`.

Input Format:

The input should be taken in via the keyboard and should contain exactly one line detailing the contact time of the two individuals.

This line contains exactly 6 space-separated integers, D1 H1 M1 D2 H2 M2 specifying when person A was in contact with person B.

D1 H1 M1 will represent the time they began contact

D2 H2 M2 will represent when the contact ended

You can safely assume that the input will always be 6 numbers separated by white spaces.

You are supposed to check that D1 and D2 are integers in the range [1:31] (including 1 and 31) telling the day of the month. Additionally, H1 and H2 are hours on a 24 hour clock with the range [0:23] (including 0 and 23), while M1 and M2 are minutes with the range [0:59] (including 0 and 59). This allows H=0, M=0 to represent midnight, while H=23, M=59 represents 11:59 PM.

If the day, hour, or minute from the input is out of range, set both contact time and risk level to -1, i.e. output `-1 -1`.

Output Format:

Write the output to the standard output, which has exactly one line containing the number of minutes that the two people were in contact with each other followed by **one white space** and the risk level of virus transmission all in **lower case**, with a **newline character** at the end of line.

Note: `System.out.println()` method prints the text on the console and the cursor remains at the start of the next line at the console (i.e. it appends a newline character at the end of the line, so you **do not** need to add a newline character by yourself if you are using `System.out.println`).

Please follow the exact output format, or otherwise you will not get any credit.

Sample Input 1:

```
java CovidTransmission
```

```
10 10 9 11 6 23
```

Input Explanation 1:

Person A initiates contact with person B on the 10th of the month at 10:09am and terminates contact with person B on the 11th of the month at 6:23am.

Sample Output 1:

```
1214 extremely high
```

Output Explanation 1:

The two people spent 1214 minutes together meaning that their contact time was > 360 mins, which puts them in the group of expremely high

Sample Input 2:

```
java CovidTransmission
```

```
10 10 9 10 10 10
```

Sample Output 2:

```
1 low
```

Sample Input 3:

```
java CovidTransmission
```

```
10 10 9 10 11 10
```

Sample Output 3:

```
61 medium
```

Sample Input 4:

```
java CovidTransmission

10 7 9 10 10 10
```

Sample Output 4:

```
181 high
```

Handling Invalid Inputs:

There are **two edge cases** that you are supposed to handle.

1. The input day is out of [1:31] range; the input hour is out of [0:23] range; and the input minute is out of [0:59] range. Output `-1 -1` to indicate invalid input. See the example input and output below
   Sample Input 5:
   ```
   java CovidTransmission
   99 9 9 9 10 10
   ```
   Sample Output 5:
   ```
   -1 -1
   ```
2. The second entered time began before the first entered time, i.e. that the two individuals spent “negative minutes” together. Obviously this is impossible and should be handled properly. If the two individuals spent “negative time” together output `-1 -1` to indicate invalid input. See the example input and output below
   Sample Input 6:
   ```
   java CovidTransmission
   10 7 9 9 10 10
   ```
   Sample Output 6:
   ```
   -1 -1
   ```

Getting Started:

If you followed the set up instructions you should have access to the command line/bash and have Java downloaded.

Create file `CovidTransmission.java` in the directory that you prefer.

After you create your file you will need to remember where it is so you can access it via bash/command line. Here are some tips for finding files via command line:

- directory means folder
- `ls` lists all files/directories in a directory

- `cd FolderName` goes into “FolderName” directory
- `cd ..` leaves a directory to previous directory

Example:

> Mia had too much fun this summer and doesn’t know where her school work is on her computer anymore.
>
> She first does `cd ~` to go to her home folder. She then does `ls` to see the following directories:
>
> ```
> Desktop Work Memes Trash
> ```
>
> Thinking her school work is in Work she does `cd Work` but then types `ls` to see the directories:
>
> ```
> FunInternship Research
> ```
>
> So she goes back a directory by doing `cd ..`
>
> Then she goes into her desktop `cd Desktop` and sees all the directories by typing `ls` where she sees:
>
> ```
> School Resumes HopesAndDreams
> ```
>
> She goes into the school folder by typing `cd School`

You can use the example as a guide or read the bash resources listed in the Setup section at the top of the assignment. Locate your file CovidTransmission.java via bash/command line and follow the instructions below.

Instructions for Compiling and running a java program after JDK installation.

- Open a command prompt window and go to the directory where you saved the java file (CovidTransmission.java).
  - if the file is in a folder on my Desktop called School I would write `cd ~/Desktop/School` in my command line
- Type ‘javac CovidTransmission.java’ and press enter to compile your code. Make sure that there are no errors in your code before proceeding to the next step.
- Now, type
  ```
  java CovidTransmission
  ```
  press ennter to run the program, then type the input

```
1 1 1 2 2 2
```

and press enter to see the output.

- You will be able to see the result of execution on the window.

Submission:

Turning in your code

Look through your program and make sure you've included your name, ID and login at the top.

Files

Submit the following file to Gradescope under the assignment "PA1":

- CovidTransmission.java

Important: Even if your code does not pass all the tests, you will still be able to submit your homework to receive partial points for the tests that you passed. **Make sure your code compiles in order to receive partial credit.**

Testing:

Try the sample inputs described above. Do you get the same results as their corresponding outputs? Now try some of your own inputs, do you get the results you would expect? If you do submit to the Autograder to see how many tests you pass.

Grading:

You can submit as many times as you want on Gradescope, and you will see the results after each submission.

Note: Only for this PA, you are able to see all the test cases and results when you submit your assignment to Gradescope. It is your responsibility to test your program comprehensively.