

**IF100 – Fall 2022-2023**  
**Take-Home Exam 1**  
**Due November 11<sup>th</sup>, 2022, Friday, 23:59 (Sharp Deadline)**

## **Introduction**

The aim of this assignment is to practice the basics of programming. You will get some inputs from the user, do some arithmetic operations and display your output to the user.

## **Description**

Let's assume that we established a company called SpaceSU to build our own space rocket (yes, we are competing with SpaceX); and we are planning a moon landing mission with our rocket. Our landing mission consists of three stages:

**(i)** Take-off: Rocket will be launched from the Earth's surface and reach the Earth's orbit. This stage will take 300 km.

**(ii)** Flight: Rocket will move from Earth's orbit to the Moon's orbit. This flight is 384400 km long.

**(iii)** Landing: When the rocket reaches the Moon's orbit, it will start the landing. The landing part is 100 km long.

We know the distances for each stage as stated above. Our aim is to estimate how long our journey will take to land on the Moon in total. To estimate the duration, we need to know about the average velocities in each stage. We will get these average velocities from the user as inputs. Unit of all inputs will be m/s (meters per second). You may assume that the user will enter inputs as integers.

## **Output**

You need to calculate the total time it will take to reach the moon with the given inputs. The output should be exactly in the following format:

*The mission will take d day(s), h hour(s), m minutes, s second(s).*

You should calculate four integers for d, h, m, s in your output. If you find one of these results as 0, you should also print that. Note that *d*, *h* and *m* values must be displayed as integers (without any floating point numbers); however *s* value must

be displayed as a real value with exactly two floating point numbers. Please check the sample runs for examples.

## Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. You have to display the required information in the same order and with the same words and characters as below.

### Sample Run 1

```
Please enter the average take-off velocity (m/s): 2500
Please enter the average flight velocity (m/s): 8000
Please enter the average landing velocity (m/s): 2000
The mission will take 0 day(s), 13 hour(s), 23 minute(s), 40.00 second(s).
```

### Sample Run 2

```
Please enter the average take-off velocity (m/s): 37
Please enter the average flight velocity (m/s): 22323
Please enter the average landing velocity (m/s): 317
The mission will take 0 day(s), 7 hour(s), 7 minute(s), 23.47 second(s).
```

### Sample Run 3

```
Please enter the average take-off velocity (m/s): 44
Please enter the average flight velocity (m/s): 4444
Please enter the average landing velocity (m/s): 44
The mission will take 1 day(s), 2 hour(s), 33 minute(s), 9.56 second(s).
```

### Sample Run 4

```
Please enter the average take-off velocity (m/s): 200
Please enter the average flight velocity (m/s): 2100
Please enter the average landing velocity (m/s): 100
The mission will take 2 day(s), 3 hour(s), 32 minute(s), 27.62 second(s).
```

## What and where to submit?

You should prepare (or at least test) your program using Python 3.x.x. We will use Python 3.x.x while testing your take-home exam. Let us repeat,

- You **must** use Google Colab to develop your code from scratch (from beginning till the end), and then submit it **through SUCourse+ only!** Once you are done with developing your code on Google Colab, then you will copy your code to CodeRunner to see if your program can produce the correct outputs. At the end, you will submit your code through CodeRunner (on SUCourse+). You should keep your Google Colab file until the end of the semester, we might want to look at this. ***If you fail to provide this Google Colab file anytime in the semester, you may not earn any credits from this Take Home Exam.***
- In CodeRunner, there are some visible and invisible (hidden) test cases. You will see your final grade (including hidden test cases) before submitting your code. Thus, it will be possible to know your THE grade before submitting your solution.
- There is no re-submission. You don't have to complete your task in one time, you can continue from where you left last time but you should not press submit before finalizing it. Therefore, you should make sure that it's your final solution version before you submit it.

## General Take-Home Exam Rules

- Successful submission is one of the requirements of the take-home exam. If, for some reason, you cannot successfully submit your take-home exam and we cannot grade it, your grade will be 0.
- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse+ time and your computer time may have 1-2 minutes differences. You need to take this time difference into consideration.
- Do NOT submit your take-home exam via email or in hardcopy! SUCourse+ is the only way that you can submit your take-home exam.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please submit your **own** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the syllabus of the course.

Good luck!  
IF100 Instructors