Program 1: Prepare for a Moon Landing...

Due date: Friday, October 17, 11:59pm.

1 Purpose

To write a program requiring use of variable declarations, reading values into and writing the values stored in variables, assignment statements and conditional statements.

Programming environment. This is a solo programming project. You are responsible for all the work related to the program development, testing and submission.

Collaboration. **Any** collaboration between peers, as well as any collaboration with outside sources is **strictly prohibited**. If you have any questions, concerning the assignment, please consult the instructor.

2 Program Description

In this project you will start implementing a version of one of the oldest known computer games: Moon Landing. In **Program 2** you will be expected to implement the entire game.

Imagine that you are piloting a one-man spaceship to the Moon. Your spaceship carries a certain amount of fuel and gives you certain *milage* for each gallon of fuel spent. Obviously you want to have some information at any moment of time about how far from the Moon you are and how much fuel you have left. The only information you have is

- Amount of fuel at the beginning of the flight
- Fuel Milage (miles/gallon)
- The velocity of your spaceship
- Distance you have to travel

Using this information you write a program that can compute how much fuel you have left and how far from the Moon you are right now (and can even give you an ETA).

There is a number of simplifying assumptions we are making here. One of them is that your spaceship travels with constant velocity. Another simplification is in the fact that we assume that the fuel milage does not depend on the velocity the ship is travelling with. We also do not take into account the loss of mass the ship is experiencing while burning fuel. Most of these problems will be handled in the actual game you are to write as Program 2)

3 Program Specifications

Write a small C program which produces several lines of output, as follows (see the sample execution in Section 5):

MLP1. The first output line shall contain the text `"Prototype Moon Landing Program."`

MLP2. The second output line shall contain your full name.

MLP3. The next few lines will contain input prompts.

MLP3-1. The first prompt shall be

```
Enter the distance to the Moon (miles):
```

The program shall read in the distance to the Moon from the starting point of the space ship.

MLP3-2. The second prompt shall be

```
Enter the initial amount of fuel (gallons):
```

The program shall read in the initial amount of fuel the ship has.

MLP3-3. The third prompt shall be

```
Enter the fuel efficiency of the spaceship (miles/gallon):
```

The program shall read in the fuel efficiency of the ship.

MLP3-4. The fourth prompt shall be

```
Enter current velocity (miles/sec):
```

The program shall read in the current velocity of the ship.

MLP3-5. The fifth and final prompt shall be

```
Enter the time in flights (mins):
```

The program shall read in the time in flight for which the computations shall be made.

MLP4. Distance to the Moon, initial amount of fuel, fuel efficiency and current velocity are floating point numbers. Time in flight is an integer.

MLP5. Each value, as it is read from input shall be tested for conformance to the rules specified below. If the value falls outside the prescribed feasible range of values, the program shall produce an error message:

```
Input Error: Incorrect <Parameter>: <Value>
```

Here, <Parameter> is one of "distance", "amount of fuel", "fuel efficiency", "velocity" and "time in flight", corresponding to the parameter that was incorrectly entered. <Value> is the incorrectly entered value.

The feasible range of values for the five parameters is defined as follows:

distance to the Moon (distance)	$0.0 \leq \text{distance} \leq 253560.00$
initial amount of fuel (fuel)	$0.0 \leq \text{fuel} \leq 40000.00$
fuel efficiency (efficiency)	$\frac{\text{distance}}{\text{fuel}} \leq \text{fuel} \leq 4 \times \frac{\text{distance}}{\text{fuel}}$
current velocity (velocity)	$7.0 \leq \text{velocity} \leq 385.9375$
time in flight (time)	$0 \leq \text{time} \leq \frac{\text{distance}}{\text{velocity}}$

Notes: 253560.00 miles is the distance from Earth to the Moon at the apogee, 7.0 miles/sec is the escape velocity of the Earth, 385.9375 miles/sec is the escape velocity of the Sun.

Note: Your program shall test each value as soon as it is read. It is considered to be a program fault for a program to read a value *out of the feasible range* and continue reading other values.

MLP6. After all the values are read, and established to be in their respective feasible ranges, your program shall compute and output the following information:

1. The distance the spaceship has travelled in the time provided to it;
2. The remaining distance to the Moon;
3. The amount of fuel used during the flight so far;
4. The amount of fuel left;
5. The estimated time of arrival to the moon at current velocity. (in hours and minutes)

For the exact format of the output, please consult Section **??**

MLP7. The last line of your program's output should contain the phrase `''I'll See You on the Dark Side Of The Moon''` (**with** the double quotes). After this line is printed, your program shall stop.

General Notes

Math. You are responsible for the remainder of the program design for this program. In particular, you are responsible for coming up with the correct math (physics) to compute the outputs of the program based on the inputs.

Program name. Name your program `travel.c`.

ANSI C. Your program shall be written in ANSI C. The instructor will compile your program using the following `gcc` flags:

```
gcc -ansi -Wall -Werror -lm
```

Any program that does not compile in this fashion will be assigned a score of 0.

Style. Your code will checked for style. Your program shall conform to the style described at

http://users.csc.calpoly.edu/∼cstaley/General/CStyle.htm

In addition, the header comment shall be as described in Lab 2 specification.

Any style violations are subject to an automatic 10% penalty.

Testing. You will be provided with the instructor's binary and with a set of test cases for testing your program. Both will be released to you at the beginning of next week (either on Monday, October 13, or on Tuesday, October 14) (a test script will also be included). Once test cases become available to you, make sure you test your code against the instructor's binary and ensure complete match of the outputs.

As usual the only dispensation is given to rounding errors due to floating point computations.

Any program that fails any of the public tests will not receive more than 30% of the grade.

4 Submission Instructions

Submission.

Files to submit. You shall submit the `travel.c` file.

No other files can be submitted. In fact, if you submit *other* file, or submit one of the three files about with an *incorrect filename*, you will receive an email informing you about a submission error, and asking you to resubmit.

Submission procedure. You will be using `handin` program to submit your work. The procedure is as follows:

- `ssh` to `vogon` (`vogon.csc.calpoly.edu`).
- Students from **Section 009** shall execute the following submission command:

  ```
  > handin dekhtyar program01-09  travel.c
  ```

- Students from **Section 011** shall execute the command:

  ```
  > handin dekhtyar program01-11 travel.c
  ```

Late submission. You may submit late for a 24-hour period following the deadline. Late submissions are subject to the standard 10%—30% penalty at the instructor's discretion.

5 Sample Output

```
> travel
Moon Landing Prototype Program
Alex Dekhtyar
Enter the distance from the starting point to the Moon (miles): 200000.00
Enter the initial amount of the fuel (gallons): 25000.00
Enter the fuel efficiency of your spaceship (miles/gallon): 20.00
Enter current velocity (miles/sec): 20.00
Enter the time in flight (mins) : 120
The spaceship has travelled 144000.00 miles
56000.00 miles left to the Moon
Fuel used: 7200.00 gallons
Fuel left: 17800.00 gallons
ETA to the Moon: 0 hours 46.666667 minutes
``I'll See You on the Dark Side Of The Moon''
```