

## CS 159 – HW #02

**Due: Monday February 15, 2021 at 11:00pm** (time local to West Lafayette, IN).

**10 Points Possible**

**Problem:** Given the mass of an airplane, the amount of forward force produced by its propellers, and the mass of the TWO gliders it is towing (airplane connected to first glider and the first glider is connected to the second glider), calculate the resulting tension on each cable connecting the aircraft and the acceleration of the glider. Vertical forces and air resistance are not considered.

- Your program must accept input and produce output that matches **exactly** to the given executions. Note that the width modifier used for the tension values is calculated as the number of digits in the force input plus six.

### Example Execution #1 (eleven spaces after colon character for tension output):

```
Enter mass of airplane (kg) -> 15000
Enter mass of glider #1 (kg) -> 5000
Enter mass of glider #2 (kg) -> 4500
Enter force produced by propellers (N) -> 75000

Acceleration: 3.06 m/s^2
-----
Resulting tension on cable #1:    29081.63 Newtons
Resulting tension on cable #2:    13775.51 Newtons
-----
```

### Example Execution #2:

```
Enter mass of airplane (kg) -> 15000
Enter mass of glider #1 (kg) -> 4500
Enter mass of glider #2 (kg) -> 5000
Enter force produced by propellers (N) -> 75000

Acceleration: 3.06 m/s^2
-----
Resulting tension on cable #1:    29081.63 Newtons
Resulting tension on cable #2:    15306.12 Newtons
-----
```

### Example Execution #3:

```
Enter mass of airplane (kg) -> 12750.9
Enter mass of glider #1 (kg) -> 2125.2
Enter mass of glider #2 (kg) -> 2075.136
Enter force produced by propellers (N) -> 38657

Acceleration: 2.28 m/s^2
-----
Resulting tension on cable #1:    9578.79 Newtons
Resulting tension on cable #2:    4732.31 Newtons
-----
```

### Example Execution #4:

```
Enter mass of airplane (kg) -> 7500
Enter mass of glider #1 (kg) -> 1260
Enter mass of glider #2 (kg) -> 1210
Enter force produced by propellers (N) -> 37500

Acceleration: 3.76 m/s^2
-----
Resulting tension on cable #1:    9290.37 Newtons
Resulting tension on cable #2:    4551.15 Newtons
-----
```

### Example Execution #5 (ten spaces after colon character for tension output):

```
Enter mass of airplane (kg) -> 775
Enter mass of glider #1 (kg) -> 120
Enter mass of glider #2 (kg) -> 100
Enter force produced by propellers (N) -> 4500

Acceleration: 4.52 m/s^2
-----
Resulting tension on cable #1:      994.97 Newtons
Resulting tension on cable #2:      452.26 Newtons
-----
```

---

**All course programming and documentation standards are in effect for this and each assignment this semester. Please review this document!**

---

### Academic Integrity Reminder:

- Please review the policies of the course as they relate to academic integrity. The assignment you submit should be your own original work. You are to be consulting only course staff regarding your specific algorithm for assistance. Collaboration is not permitted on individual homework assignments.

### Additional Requirements:

1. Add the homework assignment header file to the top of your program. A description of your program will need to be included in the assignment header. This particular header can be added to your file by entering `hhw` while in command mode in `vi`.
2. **Each of the example executions provided for your reference represents a single execution of the program.** Your program must accept input and produce output **exactly** as demonstrated in the example executions, do not add any “bonus” features not demonstrated in the example executions. Your program will be tested with the data seen in the example executions and an unknown number of additional tests making use of meaningful data.
  - All data input, output, or generated in the process of calculating the results displayed are assumed to be of the `double` data type.
3. Course standards **prohibit** the use of programming concepts beyond the material found in the first **three chapters** of the book, notes, and lectures to be acceptable for use.
  - Continue to place ALL CODE inside of the `main` function.
4. A program **MUST** compile to be considered for partial credit. The submission script will reject the submission of any file that does not successfully compile on the `guru` server. The name of the source code file you attempt to submit must be `hw02.c`, no variation is permitted.

### Course Programming and Documentation Standards Reminders:

- Indent all code found within the `main` function **exactly** two spaces.
- Place a **single space** between all operators and operands.
- Comment **all** variables to the right of each declaration. Declare only one variable per line.
- In general it is acceptable to initialize a variable declared in the local declaration section of a function. If the expression used to initialize the variable is more complex than a constant assignment then it is best to give the variable its first value inside of the executable statement section of the function.
- Notice that several programs (see program 2-9 on pages 74-75) in the programming text use a single line comment to indicate the start of the local declaration and executable statement sections of the `main` function.
  - At no point during the semester should these two sections ever overlap.
- Select **meaningful identifiers** (names) for all variables in your program.
- Do not single (or double) space the entire program, **use blank lines when appropriate**.
- There is no need to include example output with your submission.

## **Auto-Grade Tool**

- We have implemented what is being referred to as the auto-grade tool. At the time of a successful assignment submission you may receive some feedback on your program in regards to course programming and documentation standards. This feedback may include a potential deduction that you will receive once your assignment is reviewed by your grader.
- It is expected that graders verify those notes identified by this tool to ensure that they are indeed applicable and reasonable to the submission. Graders may make additional deductions for those standards not identified by the new tool.
- We hope that this feedback helps with the enforcement of course standards, consistency in grading across sections, and to encourage students to revise their work when problems are identified before the assignment deadline passes. It is possible to resubmit an assignment for grading up to the advertised deadline. Only the final successful submission is retained and evaluated.

**When you submit...** only the final successful submission is kept for grading. All other submissions are over-written and cannot be recovered. You may make multiple submissions but only the last attempt is retained and graded.

- Verify in the confirmation e-mail sent to you by the course that you have submitted the correct file (must be named hw02.c), to the correct assignment (hw02), and to the correct section.
- Leave time prior to the due date to seek assistance should you experience difficulties completing or submitting this assignment. All attempts to submit via a method other than through the guru server as set up in the Account Configuration Activity will be denied consideration.

**Assignment deadlines...** are firm and the electronic submission will disable promptly as advertised. We can only grade what you submit as expected prior to the assignment deadline.