

Assignment 2 – C Basics & Expressions



Figure 1: Queens Engineering Frosh Week. Matthew Marsh (2012)

Objective:

In this assignment, you will collect user input for the mass of the ThunderMugz components. You will use these values to calculate the number of students needed to carry the chariot. Then, you will use the values calculated to find the time needed (in minutes & seconds) to complete the ThunderMugz challenge.

Background:

ThunderMugz is one of the events as a part of Queens Engineering Orientation Week. Each Orientation Group constructs a chariot consisting of a box spring, a throne (which includes a first-year nursing student on an old toilet seat), and armor consisting of old cardboard and materials salvaged from the student housing district. Then, the ThunderMug is piloted by first-year Engineering & Nursing students through the ThunderMugz challenge, consisting of an obstacle course and a gauntlet of Thundersludge. All the teams race to see who can complete the ThunderMugz challenge the fastest.

Each group is aiming to complete the ThunderMugz challenge as fast as possible, while constructing the best armor to protect the nursing student from Thundersludge. Though for safety reasons, many students are needed to hold the heavy weight of the box spring & the throne. The number of students will affect how fast the team can complete the ThunderMugz challenge. You will create some code to determine the number of students needed to carry the chariot, as well as how fast the team completes the challenge.

APSC 143 – Introduction to Programming for Engineers

Instructions:

Below is a guideline for how to approach this problem, though you may choose to approach it however you wish. **Make sure to express your results using the format specified below:**

- Declare variables to user for user input. Ask for and collect input from the user, and store that data in these variables.
- Print an empty line.
- Use these values to calculate the total mass & number of students needed. Assume **$g = 9.81$** , and each student can carry a maximum of **300 N**.
 - Remember, you cannot have a non-whole number of students, but still enough that the ThunderMugz can be carried.
- Calculate the time needed to complete the ThunderMugz challenge using the equation below. Express this time in minutes & seconds.
 - Make sure you're using the correct libraries that allow you to use e in your code.
 - Think about what concepts you've learned so far that involve remainders, as well as rounding down intentionally to solve this problem.

$$time = mass * (4 - e^{0.1students})$$

Comments are mandatory for this assignment. Add comments as necessary for key pieces in your code, such as variable declaration, conditional statements, and looping conditions to explain what the program is doing.

Your output must match the sample output below exactly; otherwise, the auto grading software will not be able to grade your assignment, which may affect your mark.

Example Output:

(Note: You **DO NOT** need to print the values in bold; they are shown only to display the **scanf** input for this example.)

(Note: Make sure that each value is printed to 0 decimal places and that the last line also prints a new line.)

The output below displays the correct results for inputs 40 and 120, though your code must be able to accept non-integer values as well.

```
Enter the mass of the box spring in kg: 40
```

```
Enter the mass of the throne in kg: 120
```

```
The ThunderMugz will need a minimum of 6 students to carry it.
```

```
It will take 5 minutes and 48 seconds to complete the ThunderMugz challenge.
```

APSC 143 – Introduction to Programming for Engineers

Submission Instructions:

Create and submit your program to Gradescope. Do not include any personal information (student number, name, etc.) in your submission. If you have consulted any resources to complete your work, indicate those resources in your comments. Also, include in your comments that you **attest to the originality of your work**.

Refer to the assignment rubric on OnQ for a detailed breakdown of the grading criteria. Your submission must adhere to the assignment rules as outlined in the submission policy document for this course, which can also be found on OnQ. **There is zero tolerance for plagiarism in this course.** This auto grading software will automatically flag potential cases of plagiarism, which will be reviewed by the instructors.

More information on assignment submissions can be Found in Week 2, and information on the specific definition and repercussions of plagiarism can be found in the “Begin Here (About This Course)” module.