# Hack 3.0

## Computer Science I

**Department of Computer Science & Engineering**
**University of Nebraska–Lincoln**

## Introduction

Hack session activities are small weekly programming assignments intended to get you started on full programming assignments. You may complete the hack on your own, but you are *highly encouraged* to work with another student and form a hack pair. Groups larger than 2 are not allowed. However, you may discuss the problems *at a high level* with other students or groups. You may not share code directly.

If you choose to form a Hack Pair, you *must*:

1. Both join a hack pair on Canvas (go to People then Hack Pairs)
2. You must both work on the hack equally; it must be an equal effort by both partners. Do not undermine your partner's learning opportunity and do not undermine your own by allowing one parter to do all the work.
3. Turn in only one copy of the code under the individual whose last name comes first (with respect to Canvas).

You are graded based on style, documentation, design and correctness. For detail, see the general course rubric.

| Category | Point Value |
|---|---|
| Style | 2 |
| Documentation | 2 |
| Design | 5 |
| Correctness | 16 |
| **Total** | **25** |

Table 1: Rubric

Correctness: Formatting variations are perfectly fine. You need to report *just as much* information as the expected output. Points are awarded proportionally per test case.

## Problem Statement

A cell phone broadband provider sells monthly data plans to its customers. Each data plan is for 30 days and each plan gives customers a certain number of Gigabytes (GB) which must be used each month or they are lost (no "rollover"). The provider wants to help customers understand if they are using their monthly data too quickly or if they can afford to use more.

In this exercise, you will write an application for the company to help customers track their mobile data usage. Write a program that reads the following pieces of data as *command line arguments.*

- Number of GB in the plan per 30 day period
- The current day in the 30 day period (in the range 1, first day, 30 for the last day)
- The total number of GB used so far

The program should then compute whether the customer is over, under, or right on the *average daily usage* of their plan. It should also inform them of how many GB are left and how many, on average, they can use per day for the rest of the 30 day period. Of course, if they've run out of data, it should inform them of that too.

For example, if the user enters $15, 10, 13$ for each piece of data respectively, your program should print out something similar to the following.

```
10 days used, 20 days remaining
Average daily use: 1.3 GB/day

You are EXCEEDING your average daily use (0.50 GB/day).
Continuing this high usage, you'll exceed your data plan by
24 GB.

To stay below your data plan, use no more than 0.1 GB/day.
```

If the user is under their average daily use, a different message should be presented.

## Instructions

- You are encouraged to collaborate any number of students before, during, and after your scheduled hack session.

- Design at least 3 test cases *before* you begin designing or implementing your program. Test cases are input-output pairs that are known to be correct using means other than your program.
- Include the name(s) of everyone who worked together on this activity in your source file's header.
- Name your program `dataPlan.c`, and turn it in via webhandin, making sure that it runs and executes correctly in the webgrader. Each individual student will need to hand in their own copy and will receive their own individual grade.