

## Assignment #2

### Introduction to C Programming – COP 3223

#### Objectives

1. To give students practice at typing in, compiling and running simple programs.
2. To reinforce knowledge of assignment statements and arithmetic expressions
3. To learn how to use if statements for conditional execution.

#### Introduction: Mission to Mars

Your friend has been playing a new Mars Colony simulator nonstop! They are always talking about how cool it would be if they could be a on the first real-life mission to Mars! To amuse your friend, you have decided to create a series of programs about the possible first colony on Mars.

#### Problem: How Much Equipment Can We Take? (marspacking.c)

The biggest problem our mission will face is getting our supplies off Earth and on their way to Mars. Our ship has a limited amount of space and a limited amount of fuel. Most of our equipment needed to house and feed our new colony is quite large. We will need some to start with and the ability to make more later if we can find the needed resources on the Red Planet.

You will need to ask the user for the amount of space the shuttle has and for the dimensions of our most critical equipment. Additionally, ask the user for the amount of equipment we would ideally take. Then you can determine if the shuttle has enough room for everything.

If the shuttle has enough room for all the desired equipment, tell the user they can begin packing the shuttle and tell them how much space will be left over. If there will not be enough room, tell the user there is not enough room for all the equipment.

#### Input Specification

1. The amount of space will be a positive integer representing cubic feet.
2. The dimensions of the equipment will be a positive integer representing cubic feet.
3. The number of pieces of equipment will be a positive integer.

All equipment is assumed to be the same size.

#### Output Specification

There are two potential outputs. If there is enough room for all the equipment:

You have enough room for all your equipment! There is X cubic feet leftover.

If there is not enough room:

You do not have enough room for all your equipment.

### Output Sample

Below are some sample outputs of running the program. **Note that these samples are NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

#### Sample Run #1

**How much space do we have in the shuttle?**

*500*

**What are the dimensions of the equipment?**

*10*

**How much equipment will we try to take?**

*15*

**You have enough room for all your equipment! There is 350 cubic feet leftover.**

#### Sample Run #2

**How much space do we have in the shuttle?**

*400*

**What are the dimensions of the equipment?**

*20*

**How much equipment will we try to take?**

*20*

**You have enough room for all your equipment! There is 0 cubic feet leftover.**

#### Sample Run #3

**How much space do we have in the shuttle?**

*500*

**What are the dimensions of the equipment?**

*100*

**How much equipment will we try to take?**

*7*

**You do not have enough room for all your equipment.**

### Acceptable Resources

Remember, the use of online help sites is strictly prohibited. The only acceptable resources for these assignments are below:

- Course Webcourse
  - In particular: Weeks 2&3 – Arithmetic in C, Week 4 – Conditional Statements
- Course Textbook
  - Programming Knights: An Introduction to Programming in Python and C by Arup Guha
- Professor Guha's Course Archive
  - <http://www.cs.ucf.edu/~dmarino/ucf/transparency/cop3223/>
- Course TAs and Instructor Office Hours

- Getting Help: <https://webcourses.ucf.edu/courses/1336411/pages/getting-help>

### **Style Notes**

Please review the course Style Guide on the webcourse, with special attention to the following notes:

- comment major sections of code addressing: “What does this block do?” and “Why did I implement this block in this way?”
- place comments above the line(s) to which it applies
- use inline comments (//) and leave one space between // and the comment’s first character
- All variables should be declared at the top of your functions (in this program, only main is needed) and should have meaningful names
- Be sure to declare `main` with: `int main(void) {`
- Indent the contents of main four spaces or one tab
- leave a space on both sides of any binary operators you use in your code (i.e., operators that take two operands). For example, use `(a + b) - c` instead of `(a+b)-c`.
- if statements should have a single space after `if`
- contents of if statements should be indented four spaces or one tab
- if statements should not have any space immediately after each `(` or immediately before each `)`.

### **Deliverables**

One source file – *marshpacking.c* – is to be submitted over WebCourses.

### **Restrictions**

Although you may use other compilers, your program must compile and run using a standard C Development Environment. Your program should include a header comment with the following information: your name, assignment number or title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

### **Grading Details**

Your programs will be graded upon the following criteria:

- 1) Your correctness
- 2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.
- 3) Compatibility – You must submit C source files that can be compiled and executed in a standard C Development Environment. If your program does not compile, you will get a sizable deduction from your grade.