Assignment #7

Introduction to C Programming – COP 3223

Objectives
1. To learn how to write functions given specifications
2. To learn how to the use pass by value variables
3. Review previous concepts, such as if-statements and loops.

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: Building a Colony (marscolony.c)
Our shuttle of autonomous robots has landed on Mars! And now that the error in the automatic deployment software has been fixed, we can finally deploy our robots to start building our future colony.

There are five critical buildings or infrastructure elements that our colony needs to survive:
1. A habitat for humans to eventually live in
2. A hydroponic farm to provide food
3. A wind and solar farm to provide power
4. A recharging station to allow our autonomous robots to regain power
5. Water pipes to supply our habitats and farms

Each building or infrastructure element takes about one week to build. Our robots have the ability to build for three weeks without charging. After that, they will need to spend a full week recharging. We have ten weeks total to set up the colony for human habitation.

On Mars, our biggest problem will be the weather. While our autonomous robots are not affected by the cold temperatures, they will have to deal with meteor showers, wind storms, and solar flares. Some buildings and infrastructure elements can be built in adverse weather conditions, while others cannot. For example, our wind and solar farm requires perfect weather to build and set up while water pipes can be constructed in any weather. However, the intense solar flares will rob our robots of all their charge if they are building while it happens. It is not recommended to build during a solar flare unless all the elements for recharging are already in place.

If the colony is complete before the 10 weeks are up, we can stop the mission ahead of schedule and inform mission command that the colony is complete and ready for humans to arrive. If the colony is not complete by the end of 10 weeks, we will have to inform mission command that our colony is incomplete and no astronauts should be sent to Mars at this time.

Program Specification
A skeleton of the solution for this assignment is posted on the webcourse. You must fill in the functions that are currently empty. After you write each function, you should test it before moving on. The main

function should not be modified for the final submission, but you may modify it during testing, provided you return it to its initial form before submitting.

Descriptions of each function are given in the skeleton along with the function Pre- and Post-conditions. Output samples and the output specification are available on the webcourse. These show the wording you should use and how the program should run when completed. Points are allotted for following the precise wording shown.

Input Specification
The user will enter integers in response to menu prompts

Output Specification
Output to the screen. Please see the output specification text document for exact output formats.

Input/Output Sample(s)
Sample input and output files will be provided on the webcourse.

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: Week 10 – Functions (Pass by Value)
- 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:

- remove excess comments and instructions, leaving pre- and post-conditions and comments in the main function. Add comments to the supplemental functions you write as normal.
- 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 and should have meaningful names
- Indent the contents of functions four spaces or one tab
- leave a space on both sides of any binary or conditional operators you use in your code (i.e., operators that take two operands).
- keywords if, while, and for should have a single space after them
- contents of if statements and loops should be indented four spaces or one tab
- conditions should not have any space immediately after each (or immediately before each).

Deliverables
One source file: *marscolony.c* for your solution to the given problem submitted over WebCourses.

Restrictions
Although you may use other compilers, your program must compile and run using Code::Blocks. Your program should include a header comment with the following information: your name, course number, section number, assignment 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.