

Assignment #5

Introduction to C Programming – COP 3223

Objectives

1. To reinforce the use of loops.
2. To learn how to read information from files.

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: Autonomous Robot Selection (marsrobots.c)

For our first mission to Mars, we will send only autonomous robots to set up our colony. Our robot development team has recently contacted us to say that they have numerous types of robots with different attributes to choose from. We now need to decide what we will prioritize in our robots:

- Cost – this attribute specifies how expensive each unit will be
- Efficiency – this attribute specifies how long it takes each unit to complete a task
- Battery Life – this attribute specifies how long each unit's battery will last before recharging

In this program we want to read information from a file about the robots we could potentially send to Mars. Ask the user for the name of this file and process the file they specify.

For each robot, we will have a manufacturing number, a cost, an efficiency rating, and an expected battery life in hours. After we have read all the information, we want to print out which robot – identified by manufacturing number – will be best for each criterion.

- Cost – the lowest price will be considered the best
- Efficiency – the lowest rating will be considered the best
- Battery Life – the highest value will be considered the best

If a single robot has the best rating for 2 or more attributes, we should recommend it above all the other options.

Input Specification

1. The file name will be a string less than 50 characters in length

Input File Format

The first line of the input file will contain a single integer n ($5 \leq n \leq 10000$), denoting the number of potential robots, for which information is listed in the file. The following n lines will have all the information for all the robots with one robot's information on a single line. Each line will have the following format:

ManufacturingID Cost EfficiencyRating BatteryLife

ManufacturingID	will be a positive integer representing the robot type.
Cost	will be a positive real number representing the price for each unit
EfficiencyRating	will be a positive real number representing the efficiency rating
BatteryLife	will be a positive real number representing the expected battery life

Output Specification

Output to the screen. Show the user the best robot – identified by manufacturing number – for each attribute.

Attribute	Best Option	Best Value
Cost	A	X.XX
Efficiency	B	Y.YY
Battery	C	Z.ZZ

Where A is manufacturing ID of the robot with the best cost and X.XX is the best cost available, B is the manufacturing ID of the robot with the best efficiency rating and Y.YY is the best efficiency rating available, and C is the manufacturing ID of the robot with the best battery life and Z.ZZ is the best battery life available.

If a single robot has the best rating for 2 or more attributes, we should recommend it above all the other options. This is not guaranteed to happen, but is a possibility to consider:

Robot H has the best rating in 2 or more attributes!

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 7 – Files
- 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`.
- 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: *marsrobots.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.