

University of Cincinnati
Department of Electrical & Computer Engineering and Computer Science

20 ENFD 112 – Fundamentals of Programming

LABORATORY 3: PROGRAM CONTROL

Spring 2008

1. Objective

The objectives of this assignment are to practice the design of an original program without much assistance and to master the use of the `if-then-else`, `for`, `input` constructs and number to string conversion.

2. First Problem

Write a program which guesses a number between 0 and 31, inclusive, that is chosen by a user. The program proceeds by asking 5 yes/no questions of comparison. The user responds to each question by typing `y` or `n`, followed by `enter`. The program first asks whether the number is at least 16. If so, the program then asks whether the number is at least 24, otherwise whether the number is at least 8. Either way, the program eliminates half the open possibilities. That is, on the first question, the number of possibilities drops from 32 to 16. On the next question the number of possibilities drops to 8. On the next question it drops to 4. On the next it drops to 2. On the last question it drops to 1, solving the problem.

It is not hard to see that this problem can be solved using 15 `if-then-else` statements. We are asking that you solve it using 1. You can do this by defining two variables called `high` and `low`, one keeping track of the highest number the user can possibly be thinking of and the other the lowest. Initially `high` is 31 and `low` is 0. If the answer to the first question is yes, `low` becomes 16 and `high` is unchanged. If the answer to the first question is no, `high` becomes 15 and `low` is unchanged. Your program changes `low` or `high` on each question until they are equal. At that point the guess can be made. Since the question is always the same except for the values of `high` and `low`, the single `if-then-else` can be placed inside a `for` construct.

The following is a snapshot of a working program:

```
Is your number at least...16?y
Is your number at least...24?n
Is your number at least...20?y
Is your number at least...22?n
Is your number at least...21?y
Your number is...21
```

The user input consist of the `y` and `n` responses. All else is generated by the program.

3. Second Problem

The term *wind chill* was first coined by the Antarctic explorer Paul Siple in 1939 in his dissertation, *Adaptation of the Explorer to the Climate of Antarctica* where Siple proposed a formula for computing wind chill that remained in use for over 60 years. But, in the fall of 2001, the U.S. National Weather Service replaced Siple's formula with following:

$$\text{Wind chill temperature} = 35.74 + 0.6215 \cdot t - 35.75 \cdot v^{0.16} + 0.4275 \cdot t \cdot v^{0.16}.$$

where v is in the wind speed in miles per hour, and t is the temperature in degrees Fahrenheit. The old wind chill formula was:

$$\text{Wind chill temperature} = 0.0817(3.71 \cdot v^{0.5} + 5.81 - 0.25 \cdot v)(t - 91.4) + 91.4.$$

For more details see

<http://www.usatoday.com/weather/winter/windchill/wind-chill-formulas.htm>

Part I

Write a program in a file called `windchill.m` that computes and displays the wind chill from user-supplied floating point values, computed using the new formula and computed again using the old formula for comparison. Your program will prompt the user for the current temperature t in degrees Fahrenheit. If t is 30 degrees or less your program will compute the wind chill temperature using both formulas and display them rounded to the nearest one-tenth of a degree.

Part II

Add the following to your program: Display a danger message appropriate to the amount of wind chill according to the new formula. The proper danger message to display is given by the following table.

| <u>Wind Chill Temperature</u> | <u>Danger Message</u> |
|-------------------------------|---|
| Greater than 30 | Too warm for wind chill |
| 10 to 30 | Caution - No shorts allowed |
| -20 to 9 | Extreme Caution - Take a trip to Florida |
| -40 to -19 | Moderate Danger - Frostbite possible |
| Less than -40 | Extreme Danger - Don't bother leaving bed |

Part III

Add an outer loop to your program so that the user can continue to enter data, until a temperature above 100 is input whereupon the program terminates.

Part IV

Add an additional prompt at the start of the program that asks the user whether he or she wants to operate the program interactively, as above, or plot the old and new formula values as a function of both temperature and wind velocity. Provide code that displays such a plot if it is requested.

4. Help

You will have to divide integers by 2 but you will want the integer part of the result. Use `floor` to get it. For example,

```
number = floor((low+high)/2);
```

To convert an integer to a string you can use, for example:

```
s = int2str(number);
```

To convert a number to a string you can use, for example:

```
s = num2str(number);
```

To create a single string from a collection of strings use `strcat`, for example with:

```
mystr = strcat('Hello there', int2str(number));
```

To place a prompt and get string input back from the user do this:

```
user_response = input('a prompt', 's');
```

To place a prompt and get number input back from the user do this:

```
user_number = input('a prompt');
```

To display an answer, create a string, possibly using `strcat`, then apply the following:

```
disp(outstring);
```

where `outstring` is the string you want to print to the console as program output. To plot $f(x, y)$ - that means a function of x and y (for example, `wc` is a function of `t` and `v` above), first establish ranges for `t` and `v` then set a meshgrid for those variables such as the following:

```
t=-100:1:100;  
v=0:1:100;  
[t,v] = meshgrid(t,v);
```

Then write `wc` in terms of `t` and `v`. However, since `v` is a vector, use `v.^ (0.16)` and not `v^ (0.16)` when exponentiating. To visualize the result use:

```
mesh(t,v,wc);  
contour(t,v,wc);  
meshc(t,v,wc);
```

To draw to a second plot figure use:

```
figure(2);
```

To label axes and create a grid and title for the plot use:

```
grid;  
xlabel('label of the x axis');  
ylabel('label of the y axis');  
title('this is the title');
```

5. Submission

Submit `m` files for the solution to the problem of Section 2 and four solutions to parts I to IV of Section 3 on or before April 20 using blackboard. All files should be complete, working programs, not just additions to a previous part. See the course webpage at

<http://gauss.ececs.uc.edu/Courses/HTML/E112.html>

for instructions.