

Name:

Section:

Exam2: input, output, if, loops, vectors, figures.

As a reminder, no credit is given to a method that has not yet been taught. Baby steps....

I do not answer questions the first hour (unless related to machine issues).

I'm sure you've heard of the fires in California. Two teams (red and blue) are currently fighting the fire. A third team is flying in to help, and your code will tell them which team needs more help. A figure of the zones being fought displays.

```
**** California Fire Plan ****

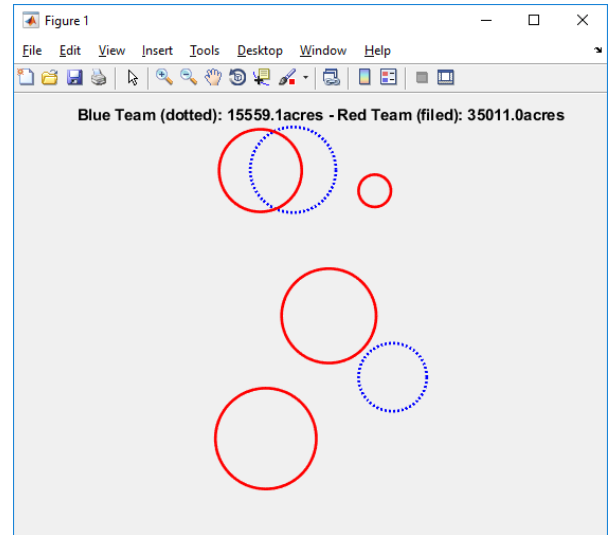
How many fires are currently burning (>4)? 4
ERROR: (whole>4)?
ERROR: (whole>4)? 9.9
ERROR: (whole>4)? 2
ERROR: (whole>4)? 6

Fire #      Fire REPORT:
          Bue Team      Red Team
1          11599.11 acres
2          9526.25 acres
3          1355.02 acres
4          6032.88 acres
5          13203.22 acres
6          8853.65 acres

----- TOTAL -----
          15559.13 acres  35011.01 acres

Team3 - go help team red!!!
```

← A sample run



### Code Requirements:

An intro message is always required.

The number of fires is a user input. **Trap** while invalid. What is considered invalid is shown in the sample run above.

Each fire is represented by a circle. The parametric equation to plot a circle is:

$$x(\theta) = radius * \cos(\theta) + x_{center}$$

$$y(\theta) = radius * \sin(\theta) + y_{center}$$

Where  $\theta$  goes from 0 to 360 degrees (to plot a full circle).

The values  $radius$ ,  $x_{center}$  and  $y_{center}$  are randomly generated by MATLAB for each fire:  $radius$  is a **float** between 20 and 70, and  $x_{center}$  and  $y_{center}$  are random **integers** between -200 and 200.

Calculate the areas for each fire (in acres) so you can add them up according to teams.

Which team is fighting the fire is also determined by MATLAB each time: generate a random **integer** 1 or 2. As the programmer, you decide: one value means Red team, the other means Blue team. Each time MATLAB generates the data, plot the circle in the blue/red color and keep two separate sums of the areas. (Remember to use `hold on`;) )

To determine which team the third team should go help, use the following factor.

$$factor = \frac{sum\ of\ areas\ of\ blue\ team}{sum\ of\ areas\ of\ red\ team}$$

If  $factor > 1.5$  team3 should help the blue team, if the  $factor < 0.5$ , go help the red team. Anything in between means team3 can help whomever they want. One of those 3 options should display.

### Requirements for the command window display:

It is crucial the output looks professional and spaced out as in all our labs.

1) The fire's number (1,2,3,... Etc) must show.

- 2) A clear visual of which team is fighting that fire, with the area displays. (You do not have to match my sample run, but it must be OBVIOUS and somewhat a table looking output).
- 3) However you decide to show the values, the values must align at the decimal point.
- 4) Units are required everywhere. All values show the same amount of decimals (your choice, at least >1).
- 5) Once calculated, the total acres for each team must also show (At your preference where and how.)
- 6) Where team 3 should go must display.

**Requirements for the plot display:** Use the plot() command learned week3/4. Create a dynamic title that shows the total for each team and which linetype. (The legend() won't work for this. Don't try.)

#### Check list – read before starting

##### The basic:

- 5pts Valid Filename (up to you)
- 10pts Code runs. Comment out what crashes but leave it!
- 6pts Name, section, valid description
- 2pts each Necessary clean up commands
- 10pts Completed algorithm as comments – requires no programming knowledge. Just English.**
- 10pts Descriptive variable names. NO single letters except for the loop control variable.
- 10pts Space out the code as shown in all examples/solutions. Be consistent.
- 5pts Proper use of semi-colons. (0 or 5pts. 1 rule to know)
- 5pts A valid intro message
- 10pts Human factor: space out the output. Lots of \n\n, tabs, spaces, field widths.

##### The solution:

- 5pts Ask user for quantity of fires
- 10pts Trap for all invalid options
- 10pts Loop to display each fire data (loop of your choice)
- 15pts Generate random radius, x and y center
- 5pts Calculate area for each circle
- 5pts Generate random team number
- 10pts Valid displays, numbers align by teams
- 10pts Valid plot with color according to team, all on same plot, no axis
- 10pts Adding up the areas by team
- 5pts Display the total areas for each team
- 5pts Determine factor
- 10pts Correct display of where team3 should go
- 5pts Dynamic title shows the correct data
- 10pts Leeway

##### The engineer:

- Step7c. Twice. Located at bottom of the script file. If something looks off, write a note!
- 10pts 1 test must show the loop traps ALL invalid values in one run.
- 10pts Another test for the rest of the code

Do not wait until the code runs to do this. Even fully testing your trap loop is 1 valid test. Crash tests are ok too if towards the end of the time (then comment what crashes before submitting).