

Problem Set #3

Assignment Learning Objectives

1. Write a program that uses selective and repetitive control to solve higher complexity programming problems and reads input from a file
2. Demonstrate efficient coding practices using multi-dimensional arrays
3. Develop code to calculate common meteorological/climatological values
4. Practice robust computer coding strategies

Due 18 September 2020 by 4:00 pm

Due Online

- The script in the PSet3 repository on your GitHub account
- A readme.txt file that explains what your script does and how it works

Due Via Blackboard

- A PDF version of your code. Make sure the honor code appears in the comment block at the beginning of the script and that it has your full typed out name. This will serve as your assertion that you have upheld the honor code.
- A PDF version of the readme.txt file

Honor Code

You must type the full honor code into each comment block to indicate that you have upheld the code. Authorized aid for this assignment is your notes and any previous programs you have written. You are highly encouraged to work with others in the class to help diagnose bugs and work out programming logic. Any copying of someone else's code IS A VIOLATION OF THE HONOR CODE, whether from a classmate, or the Internet. *Be sure to indicate with whom you have worked in the comment block of your submission.*

Background

Often in meteorology the total value of an atmospheric variable is not as informative as its anomalous value. In reality it is the anomaly (or perturbation) from a mean state that is truly driving weather and climate systems. The total value of a variable can be defined as the sum of the mean and an anomalous quantity

$$a = \bar{a} + a'$$

where \bar{a} is the mean value of a and a' is the anomalous value. To compute the anomaly the above equation can be easily solved by algebra, yielding

$$a' = a - \bar{a}$$

There can be any number of different anomaly values depending on how the average is defined. In the case of global climate change, the average temperature is taken to be a period of thirty years (currently 1981-2010),

whereas you could define the model domain anomaly by taken the average of the whole domain to determine local maximum and minimum anomaly values.

Problem

1. Write a program to read a series of surface temperature values in Kelvin that cover the Midwest, compute the anomaly values over the entire domain, report the mean temperature of the domain, determine which point contains the warmest and coldest anomaly values, and have the program generate warnings for anomaly values greater than (less than) 10 F (-10 F). Each warning should have its latitude/longitude position associated with it and a final count of the number of high and low warnings should also be reported. Output temperatures should be in Fahrenheit.

Additional Information

- The grid contains 728 points in a 28x26 (lon [col] by lat [row]) box. The upper left latitude of the domain is 55 N and the upper left longitude is -100 E (100 W). The grid spacing is 1 degree in the North-South direction and 2 degrees in the East-West direction.
 - Warning for anomaly values greater than 10 F: "Excessive Heat Warning"
 - Warning for anomaly values less than -10 F: "Extreme Cold Warning"
 - An example input file has been included in your GitHub repo for this assignment
 - Latitude and Longitude files are also included for reference and don't have to be used
 - Using Fortran formatting statements will help create high quality output, but is not required
2. Include a readme.txt file in your GitHub repository that explains how the script works, how to run it, and what output will be generated.
 - A brief description of the project
 - Use instructions
 - A short example/tutorial

Program Name: **anomaly_warning.for**

Program Input: **surface_temps.dat**

Program Output:

- To standard output
- The mean of the surface temperatures
- Warnings for Individual grid points, as necessary with the anomaly value
- Total number of warnings by type
- Warmest Anomaly with associated lat/lon point
- Coldest Anomaly with associated lat/lon point

- To file: **sfc_anomaly.dat**
 - Calculated anomaly values in Fahrenheit

Notes:

- Break the problem down into smaller bite size pieces. For example, first write your program to be able to read in the input file and compute the average over the entire domain. Then submit that working code to your GitHub repository for this assignment. Now go back and convert the temperatures to Fahrenheit, and so on.
- Make sure to regularly submit your work to your GitHub site. Part of your grade on each assignment is on your regular submission of the program(s) as they are in progress of being developed. This is easily done when regularly submitting the bite size pieces that you create when breaking down any coding project!
- Test more than just the cases or example file given below. An important part of code development is testing to make sure your code works for ALL cases, so think about "edge" cases that you know to be problematic and test those specifically.

Please make your program to produce output in the same fashion as the test cases below.

Test Case

>>>

Extreme Cold Warning for 55 N -74 E

Anomaly value: -10.294319 F

Extreme Cold Warning for 55 N -72 E

Anomaly value: -16.432327 F

.

.

.

.

Excessive Heat Warning for 30 N -80 E

Anomaly value: 10.297684 F

Number of Warm Warnings: 75

Number of Cold Warnings: 125

Average Temp: 70.6623383 F

Warmest Point: 45 N -98 E

Warmest Anomaly: 17.353653 F

Coldest Point: 54 N -64 E

Coldest Anomaly: -33.496346 F

>>>

Evaluation Criteria

Each of the following criteria will be rated from not present/completed to exemplary, having all of the elements will yield at least a 7/10 for a particular criterion. The assignment is out of 50 points.

- Mean temperature is correctly and efficiently coded, determined, styled well, and reported in output (10 points)
- Efficiently coded, styled well, and correct reporting of Warmest/Coldest point and value (10 points)
- Efficiently coded, styled well, and correct warnings issued including total number of warnings (10 points)
- Informative and Clear Output from compiling and running code (10 points)
- Code is well documented (informative comments/descriptions of code blocks) and readme.txt file is good (5 points)
- Code was regularly updated and committed to GitHub repository (5 points)