

CEG 2170

Programming Assignment 5

Purpose of Programming Assignment 4

The primary purpose of this assignment is to gain experience with input files and arrays, primarily parallel arrays. You will use parallel arrays to store information about hurricanes (based on hurricane category) and use that information to determine which storms align to which hurricane category, if any. Some storms are just that – storms; they are not classified as hurricanes (e.g., tropical storm). The hurricane category and storm data are read from separate input files.

Overview

The Saffir-Simpson scale defines categories of hurricane intensity based on a storm's wind speed. This scale of hurricane intensities is used to classify hurricanes according to the amount of damage that a storm is likely to generate if it hits a populated area. The main characteristics of the five categories are described below:

Category	Wind Speeds (mph)	Storm Surge (ft)	Damage
1	74-95	4-5	Minimal damage to property
2	96-110	6-8	Moderate damage to property
3	111-130	9-12	Extensive damage to property
4	131-155	13-18	Extreme damage to property
5	155+	18+	Catastrophic damage to property

You will read the separate input files for the hurricane categories and storm data into separate sets of parallel arrays. The categories, minimum wind speeds, and maximum wind speeds will be read (in that order) from one input file into a set of three parallel arrays. The storm ID and storm peak wind speed (in that order) will be read from a second input file into a set of two parallel arrays. You'll use the information in the storm arrays to categorize each as a hurricane, if applicable. Those storms that are not hurricanes are skipped in the tabular output. Finally, while processing each storm, you'll track which storm is the least powerful and the most powerful (as determined using the peak wind speed).

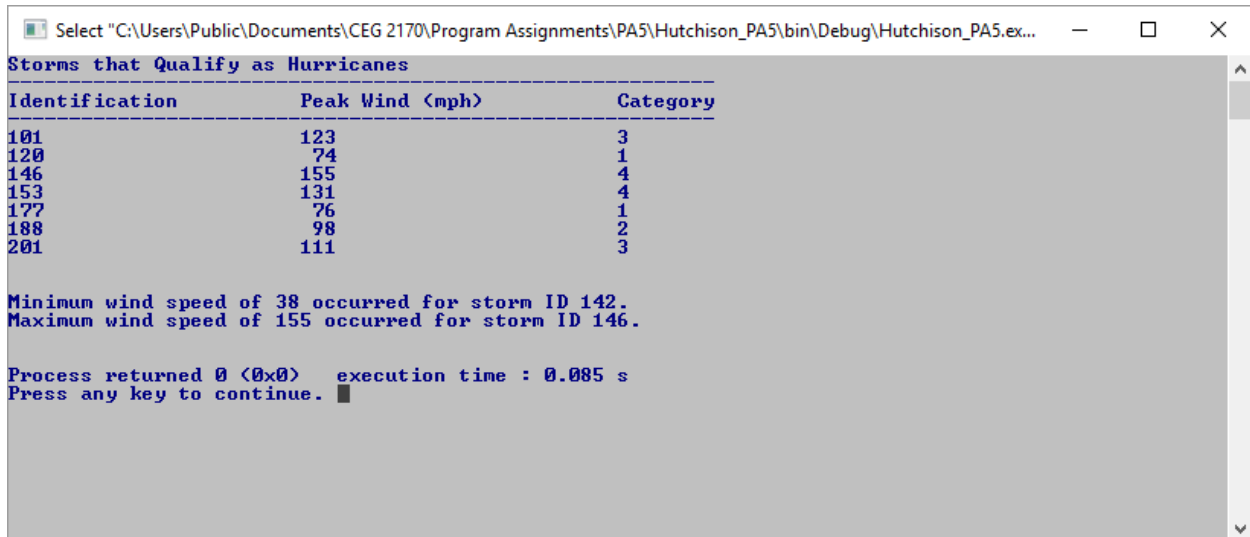
Requirements

1. In `main`, create three separate arrays to store the three sets of information from the hurricanes input file and two separate arrays to store the two sets of information from the storms input file. You are required to pass these arrays to the other functions. The remainder of `main` consists simply of four function calls to `readCategoryInfo`, `readStormInfo`, `printHeader`, and `processStorms`.
2. The `readCategoryInfo` function accepts the three hurricane information arrays as arguments and returns nothing. This function will simply read the hurricane information from the `categoryInfo.txt` file into these three arrays. Error checking is required when opening the file; exit the program if an error occurs reading the file.
3. The `readStormInfo` function accepts the two storm information arrays as arguments and returns nothing. This function will simply read the storm information from the `storms.txt` file into these

two arrays. Error checking is required when opening the file; exit the program if an error occurs reading the file.

4. The `printHeader` function simply outputs the table header in the same format shown below in the Sample Output.
5. The `processStorms` function accepts all five arrays as arguments and returns nothing. This function will determine whether each storm is categorized as a hurricane. If so, output the storm's ID, its peak wind speed, and its hurricane category. You are also required to determine the weakest and most powerful storm in the input file.

Sample Output



```
Select "C:\Users\Public\Documents\CEG 2170\Program Assignments\PA5\Hutchison_PA5\bin\Debug\Hutchison_PA5.ex..."
Storms that Qualify as Hurricanes
+-----+-----+-----+
Identification    Peak Wind (mph)    Category
+-----+-----+-----+
101               123                3
120               74                 1
146               155                4
153               131                4
177               76                 1
188               98                 2
201               111                3

Minimum wind speed of 38 occurred for storm ID 142.
Maximum wind speed of 155 occurred for storm ID 146.

Process returned 0 (0x0)   execution time : 0.085 s
Press any key to continue.
```

What to Turn In

Upload your zipped (compressed) project to the Dropbox on Pilot. Be sure your program follows the guidelines given on the Style Requirements document (provided on Pilot) with respect to commenting, variable naming, indenting, etc.

Grading Rubric

General documentation/style (10 points possible; add all that apply):
Complete header information at top of program (2 pts)
Correct indenting (2 points)
Correct use of white space; in general, code is single-spaced with a blank line between sections (such as between variable declarations and input; between input and calculations; between calculations and output). This applies to main and all other functions (3 pts)
Clean and organized presentation of your program input/output presented to the user (3 pts)
Main (22 points)
Array declarations (10 pts)
Function calls for readCategoryInfo, readStormInfo, printHeader, and processStorms (12 pts)
Functions (53 points)
readCategoryInfo with error checking (12 pts)
readStormInfo with error checking (12 pts)
printHeader (5 pts)
processStorms (24 pts) <ul style="list-style-type: none">• Parallel array processing (16 pts)• Minimum storm processing (4 pts)• Maximum storm processing (4 pts)
Program Miscellaneous (15 points)
Include proper include statements (3 pts)
Meaningful variable names (5 pts)
Proper use of required constants (2 pts)
Proper specification of function prototypes (5 pts)