

# CS-102 Project 4 – Pokémon Organizer

Professor: Brian Frost

Summer 2019

## Specifications

In class, we looked at a program that computed a student's average grade and then sorted the students in alphabetical order. In this assignment, you will follow a similar procedure, but with more options and with more complicated structures.

Your structures are based on video game characters from the series Pokémon, one of my favorite series! In the first Pokémon games, there are 151 unique species of Pokémon, each with a name, a 5 base stats, and either one type or two types. I have given you a text file which lists all Pokémon in the following format:

NAME HP ATTACK DEFENSE SPECIAL SPEED TYPE1 TYPE2

Where TYPE2 is the word NONE if the Pokémon has only one type. Each team in Pokémon is composed of 6 Pokémon, and a common challenge in Pokémon games is to use only Pokémon of a certain type. Furthermore, a common metric for evaluating the strength of a species is the sum of its five base stats.

The basic functionality of the program should be as such – I should be prompted to enter a type. If I enter a valid type, I should be told the six species which have the highest sum of stats. If there are fewer than 6 valid species (for example, there are only 3 DRAGON Pokémon) I should be told however many there are. The program should also create a file ranking all species of that type. The file should just have the species name and the sum of their stats. If there are ties, any order can be chosen. I should be able to choose this file's name.

Lastly, if when prompted for a type I type 'all', the output file should rank all 151 species. I should be told the 6 species with the largest sum of stats out of all of these.

The output files should be formatted as **NAME STATSUM** where the name is left-justified and constrained to ten characters and the stat sums are aligned in a column (using a tab character).

My suggestion is that you rely on the format of the program shown in class as an example. It will be very helpful, so long as you have the insight to adapt the process to this case. For example, you can use the same sort of sorting algorithm, but you are sorting based on a different variable.

An example of operation:

```
Enter Type: ground
Enter name of output file: ground.txt
The following Pokemon belong on your team:
RHYDON
GOLEM
NIDOQUEEN
NIDOKING
SANDSLASH
DUGTRIO
```

Then, the file `ground.txt` will have the following contents:

RHYDON	440
GOLEM	420
NIDOQUEEN	410
NIDOKING	410
SANDSLASH	405
DUGTRIO	355
GRAVELER	345
MAROWAK	345
ONIX	340
RHYHORN	315
SANDSHREW	280
CUBONE	270
GEODUDE	270
DIGLETT	230

## Submission and Grading

Please submit your assignment as a `.c` file to `b.frost@columbia.edu` by midnight on Wednesday, August 7. Remember that lateness is penalized! Please state which operating system, text editor and compiler you used in your email. Also, please don't hesitate to ask as many questions as you need through email, or by coming to office hours.

Grading is broken down according to **Correctness** and **Style**. Your total grade is out of 100 points.

**Correctness (80 points):** This is simply how well your program fits the exact above-stated specifications.

**Style (20 points):** This is composed of efficient commenting, intuitive spacing and indentation, intuitive variable names, and overall elegant code. Unnecessary statements, poor placement of variable instantiations and otherwise difficult-to-read code will result in a loss of points here. Remember to put a brief comment at the top of your code explaining the program's operation.