

## CS 410: Homework 2

**Due: Tuesday, Nov. 5, 2019 at noon**

**100 pts**

### Instructions:

**YOU WILL SUBMIT MULTIPLE FILES FOR THIS PROGRAMMING PROJECT.**

Create a subdirectory named "hw2" in your cs410 directory. Use that subdirectory for all your file submissions on this assignment. At the end of the homework assignment, as a minimum, these files should be found in your hw2 directory:

- **hw2\_functions.cpp** (function definitions)
- **hw2\_functions.h** (function declarations)
- **hw2.cpp** (main)
- a "typescript" file demonstrating program compilation, execution and testing

You may include other source/header files if you please.

**Background:** After weeks of stress, Krusty is taking a vacation to Atlantis. But now, the Atlantians (those wierdos living in Atlantis) have made it so that he can't leave to go back home unless he jumps through the right hoops. Those "hoops" are that he first visits with enough of the inhabitants (weirdos) to satisfy the Atlantis Visitors' Bureau. And to prove that he has done so, he has to collect the business cards of each one he visits and show the "gate keepers" as he leaves. There's just one problem ..... Krusty is his own worst enemy in that he has a business card fetish: he won't carry in his hands multiple cards unless he can stack them in such a way that no card is smaller than the card below it<sup>1</sup>. Now, he can orient the stacked cards any way he wants, but a card can't "hang over" (or be the same size as) the card above it. This is stupid, but oh well! It's part of the reason why Krusty is an interesting character. Fortunately for Krusty, there's an out: IF he visits enough *different* inhabitants, then he can get out of Atlantis regardless of how the cards are stacked.



**Specifications:** Your program is to simulate this situation. Krusty will visit **at random** some of the various "creatures" (inhabitants) of Atlantis, whereupon each of the creatures will say something chosen at random from a list of comments (below), and then give him their card which he will put in his pocket. Each creature has a card specific to them. What is important about their card is its dimensions: length and width (assume these are integers). Each time he gets a card, he needs to assess whether or not he can leave Atlantis (like really! No good hay-

---

<sup>1</sup> As an example: card1 is 4 x 8, card2 is 2 x 9. These can NOT satisfy Krusty's criteria since there is no way to orient (at right angles) these two cards without one hanging over the other. As another example: card1 is 2 x 3 and card2 is 1 x 2. These two CAN satisfy Krusty's criteria since card2 can be stacked below card1 in a way that it will not hang over the top card since  $1 < 2$  and  $2 < 3$ . So, you see, you must compare the cards in both orientations. Both dimensions of a lower card must not exceed either dimensions of an upper card, and at least one dimension of an upper card must be strictly greater than at least one dimension of a lower card. (Another ex: a 1 x 1 card can be below a 1 x 2 card)

bars...only sushi restaurants.) When he confirms he has satisfied the Rules of Leavature, he needs to output an exclamation that he is leaving (we'll leave the wording up to you - make it clean) and returning home, and state with detail what gives him the right to leave. Leavature Rules are specified below. And what details are we expecting? Either output the names of the creatures' cards and their dimension of the 6 stackable cards<sup>1</sup>, or output the 14 different names of the 14 different creatures he got cards from.

### Details:

- You are expected to use a struct for a creature that includes its name and the dimensions of its card.
- You are expected to use vectors (**NOT arrays**) in this assignment as appropriate. For example, you could set up a const string vector loaded with the sayings that the creatures can speak. Think about how you could randomly choose an entry from that vector.
- You can assume that our hero Krusty will collect no more than 150 cards. (Hmmm, is that another hint about an appropriate use of vectors???)
- Leavature Rules (a.k.a. Rules of Leavature): Krusty can leave if he has 6 (six) "stackable"<sup>1</sup> cards (like, you know, n'stuff, non-overlapping); OR he has cards from 14 **different** creatures (implies 11 tunas' cards and 3 sardines' cards ain't gonna do it!)
- In your code, generate a random creature before you generate a random saying.
- If you end up stacking cards that you had repeats of (e.g. you stack a 2 x 3 and you got two of these cards from two different creatures), we care not which creature's card you use.
- Each creature has many (infinite number of) cards. So, each time it is visited, it can hand a card to Krusty
- The creatures (First\_name Last\_name) and their card specs are listed in the file **creatures.txt**. E.g. Wanda Walrus 2 x 6 means First\_name = Wanda, Last\_name = Walrus, card specs = 2 x 6.
- The things the creatures can say are also listed in the file **things.txt**.

Assuming that Krusty walked for a distance of 6 miles before satisfying the rules of leavature, write a function that accepts weight, height and temperature as arguments and uses a lamda function to calculate and display the Caloric intake for Krusty according to the formula:

$$\text{Cals} = 6 * \text{wt} + 2 * \text{ht} + 1 / \text{temp} + \text{distance}(\text{in miles})$$

**Special Underwater Observation:** You will find your coding easier if you were to sort your cards when gathering them. Think carefully how this has to be done. You wouldn't want to *overload* your day with an *overload* of work when *overloading* your brain while coding...

**When you submit:** seed your random number generator with 37. Please understand that we specify the seed for submission on the hopes that your output will be the same as everyone else. However, because of slight differences in how you all code, this may not work out as planned. But, we're hoping that it pays off to some significant degree.

And, as always, let your TAs or instructor know if you need any help.