

CS 410: Homework 6

Due: Tuesday, Oct. 25, 2018 at noon

100 pts

Instructions:

YOU WILL SUBMIT MULTIPLE FILES FOR THIS PROGRAMMING PROJECT.

Create a subdirectory named "hw6" in your cs410 directory. Use that subdirectory for your all file submissions on this assignment. At the end of the homework assignment, as a minimum, these files should be found in your hw6 directory:

- **hw6_functions.cpp** (function definitions)
- **hw6_functions.h** (function declarations)
- **hw6.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, Hans Moleman 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 to 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 Hans Moleman 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 it¹. 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 Hans Moleman is an interesting character. Fortunately for Hans Moleman, 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. Hans Moleman 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-bars...only sushi restaurants.) When he confirms he has satisfied the Rules of Leavature, he needs to output

¹ As an example: card1 is 4 x 8, card2 is 2 x 9. These can NOT satisfy Hans Moleman'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 Hans Moleman'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)

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 cards¹, 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 arrays in this assignment as appropriate. For example, you could set up a const string array loaded with the sayings that the creatures can speak. Think about how you could randomly choose an entry from that array.
- You can assume that our hero Hans will collect no more than 150 cards. (Hmmmm, is that another hint about an appropriate use of arrays???)
- Leavature Rules (a.k.a. Rules of Leavature): Hans can leave if he has 6 (six) "stackable"¹ 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 cards. So, each time it is visited, it can hand a card to Hans
- Here's the data to use (creature and their card specs):
 - Wanda Walrus 2 x 6
 - Stanley Sardine 3 x 1
 - Sylvia Seahorse 4 x 2
 - Janie Jellyfish 1 x 10
 - Doris Dolphin 8 x 4
 - Bob Blobfish 1 x 5
 - Sammy Shark 8 x 4
 - Walter Whale 6 x 2
 - Stevie Salmon 2 x 3
 - Sheila Shellfish 1 x 3
 - Daniel Octopus 3 x 7
 - Mark Herrings 9 x 5
 - Bernie Tuna 3 x 5
 - Oscar Tilapia 5 x 3
- Here's a list of the things they can say:
 - Whazzup?
 - Duuude, totally love the horse head and human body combo!"
 - Looking for a card (shark)?
 - Are you someone famous?
 - You look fishy to me; or are you horsing around?

- Don't trust the salmon!!!
- Here's some advice: the shark wants you to be his chum!
- Better take some paper towels if you visit the blobfish
- Hey man, got any sea-weed?
- Let me give you my card...my phone number is on the back
- I'd gladly pay you Tuesday for a lobster roll today.
- Don't be so crabby!
- Just keep swimming, swimmin, swimming, ...
- You just keep sinking, sinking, sinking,...

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.