Genomics (Ecol 553) Computational Lab

Week 6: Sept 25 & 27, 2011.

Course webpage: <http://genomics.arizona.edu/553/>Lab

Topics: Perl lists and arrays

In class exercises:

1. Copy the directory /genome/student/ecol553\_week6 to your home directory on HPC
2. Run the array\_sort.pl script. Then modify the script to add the name of your favorite band to the *beginning* of the @arr array *using a Perl function*. (Add one line of code after the push).
3. Run the modified array\_sort.pl. What position is your band’s name in when the array is sorted?
4. Open flow\_demo.pl with an editor and look at the first 12 lines. What do you think the ‘exit’ statement does? Run the script and decide whether the first 12 lines are correctly written (if you need help, remove the comment character # from the use warnings line.)
5. In flow\_demo.pl, comment out the exit on line 12 and look at lines 14-25. Add the code indicated by the comment on line 18 and re-run the script to test it.
6. In flow\_demo.pl, comment out the exit on line 25 and look at the lines below that. Run the script and check to make sure you understand foreach, next, and last. Change the first “if” test to check for different numbers to skip and change the second “if” to jump out of the loop at a different number. Test your changes.
7. Look at hash\_demo.pl and run it. Make sure you understand the code, as it is the basis for homework problem #3.

Homework. To be completed by 11:59pm on Tuesday, Oct 2.

1. A Fibonacci Number is a location in the Fibonacci sequence defined as

Meaning the n-th number is simply the sum of the previous two Fibonacci numbers. This can be done simply using Dynamic Programming. If an array @Fib contains all of the Fibbonacci numbers up to n-1, we can simply calculate the n-th number.

Create a script hw4\_1.pl so that at run time it looks as follows:

 >./hw4\_1.pl 7
 **The 7-th Fibonacci Number Is: 13**

Here the number comes from the command line, and finds that Fibonacci number.
2. Using chomp\_defined.pl as a starting point, rename the script as name\_age.pl

Make the following modifications to the script:

a. Comment out the if(!defined $age) code block

b. Add code to the script so that it prompts the user to enter his or her age.

c. If the age is greater than or equal to 50, print “Your age should be kept a secret!!\n”, otherwise print the age as “You are \_\_\_\_ years young.\n” (*Replace the \_\_\_ with the user’s age)*

1. Using foreach\_array.pl as a starting point, rename the script as foreach\_subtract.pl. Make the following modifications to the script, and don’t forget to correct the comments!
	1. Change the initial range of the array to be 21 .. 42
	2. Use the push function to add the values 99 and then 103 to the end of the array.
	3. Change foreach loop so that it subtracts 2 from each array element instead of adding 1.
	4. At the bottom of the script, add a second foreach loop that multiplies each (previously modified) array element by 5.
	5. After each loop, print the resulting array values.
2. Using hash\_demo.pl as a starting point, rename the script as hash\_bookauth.pl

Make the following modifications to the script:

* 1. Before the code that sorts the library hash, add code to prompt the user to input the title and the author of their favorite book. Don’t forget to use chomp() to remove newlines from both. Store these values in the library hash, using the book title as a hash key.
	2. In the final foreach loop, add variables $max\_books and $max\_auth to determine which author in the library has written the most books. Initialize $max\_books to zero before the loop, and initialize $max\_auth to the empty string “”. Inside the loop, compare $bookcount{$a} with $max\_books and update $max\_books and $max\_auth if necessary. After the loop, print the name of the author with the most books in the library, and the number of books for that author. (Ignore the case where there is a tie.)