Homework #4

Due: Wednesday, November 28, 2012 at 5:00PM **Points:** 100

Please turn in your answers for the homework assignment on SmartSite, under Homework #3 in Assignments. Turn in your programs answers for the extra credit under Extra Credit #3 there.

1. (*30 points*) A string is said to be *abcdearian* if the letters in it, regardless of case, are in dictionary order. So, for example, "almost" and 'effort" are abcdearian, and "willow" and "computer" are not.

   (a) Write a *recursive* function called `isabcde(s)` that returns `True` if *s* contains a string that is abcdearian, and `False` otherwise. The function must ignore any non-letter characters in *s*, and treat all alphabetic characters as lower case.

   (b) Write a program that reads a string and uses the function to determine whether the string is abcdearian. The program is to loop until the user types an end of file (control-D), or another exception occurs.

   Here is sample output:

   ```
   The string? heLlo↵
   heLlo is not abcdearian
   The string? aLmost↵
   aLmost is abcdearian
   The string? w3i$l0l!ow↵
   w3i$l0l!ow is not abcdearian
   The string? e3f$f0o!rt↵
   e3f$f0o!rt is abcdearian
   The string? cOmpuTer↵
   cOmpuTer is not abcdearian
   The string? ABcDE↵
   ABcDE is abcdearian
   The string? control-D
   ```

   Please call your program "abcde.py".

2. (*70 points*) The *birthday problem* asks how many people must be in a room so that the probability of two of them having the same birthday is 0.5. This problem has you explore it by simulation. Basically, you will create a series of lists of random numbers of length $n = 2, \ldots$, and look for duplicates. You will do this 5000 times for each length. For each length, count the number of lists with at least 1 duplicate number; then divide that number by 5000. That is the (simulated) probability that a list of n generated numbers has at least one duplicate. As the random numbers you generate are between 1 and 365 (each one corresponding to a day of the year), this simulates the birthday problem.

   Now, breathe deeply and calm down. We will do this in steps!

   (a) First, detecting duplicates. Write a function called `hasduplicates(l)` that takes a list *l* and returns `True` if it contains a duplicate element, and `False` if it does not. For example:

   ```
   >>> hasduplicates([1, 2, 3, 4, 5, 5, 2])↵
   True
   >>> hasduplicates([1, 2, 3, 4, 5, 6, 7])↵
   False
   ```

   (b) Now, deal with one set of birthdays. Write a function called `onetest(count)` that generates a list of *count* random integers between 1 and 365 inclusive, and returns `True` if it contains a duplicate element, and `False` if it does not. Please use the function `hasduplicates(l)` to test for duplicates.

   Hint: To generate a random number between a and b inclusive, put

```
import random
```

at the top of the program, and then call the function `random.randint(a, b)`.

(c) Now for the probability for *count* people. Write a function `probab(count, num)` that runs *num* tests of *count* people, and counts the number of tests with duplicates. It returns the fraction of the tests with duplicates; that is, the number of duplicates divided by *num*.

(d) Now for the demonstration. Start with 2 people, and begin adding people until the probability of that many people having two people with a birthday in common is over 0.9. (In other words, start with a list of 2 elements, and increase the number of elements in the list until the simulation shows a probability of 0.9 that a number in the list is duplicated.) Print each probability; your output should look like this:

```
For  2 people, the probability of 2 birthdays is 0.00220
For  3 people, the probability of 2 birthdays is 0.00880
For  4 people, the probability of 2 birthdays is 0.01680
For  5 people, the probability of 2 birthdays is 0.02940
For  6 people, the probability of 2 birthdays is 0.03940
For  7 people, the probability of 2 birthdays is 0.05900
For  8 people, the probability of 2 birthdays is 0.06840
For  9 people, the probability of 2 birthdays is 0.09700
For 10 people, the probability of 2 birthdays is 0.12360
```

How many people are needed so that the probability of two of them with a birthday in common is over 0.9? How many are needed such that the probability of two of them having the same birthday is at least 0.5? Put these answers into a comment at the head of the file.

Hint: Don't be surprised if your probabilities are slightly different than the ones shown in the sample output. As randomness is involved, it is very unlikely your numbers will match the ones shown here.

Extra Credit

Remember to hand this in under Extra Credit #4, *not* under Homework #4!

3. (*30 points*) Ackermann's function $A(m, n)$ is defined as follows:

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

(a) Write a function named `ack(m, n)` that evaluates Ackermann's function. Then write a program that calls that function after reading m and n. (Remember to check that they are nonnegative integers!) Test your program on $A(3, 4) = 125$.

(b) For $m = 4$ and $n = 4$, how many calls to `ack` are made before the program terminates? How does it terminate?