UPL Interview Prep

October 1, 2015

Complete the following questions as best as you can. Write out the answer on paper (without looking anything up), and then attempt to actually write your code at the HackerRank link with any resources you would like.

These are some common problems I've encountered interviewing with various companies (Microsoft, Google, Dropbox ... to name a few) as well as in Cracking the Coding Interview.

1 Getting Started

Write a function that takes a number N. For all integers between 1 and N, if the integer is a multiple of 3, print "Fizz". If it is a multiple of 5, print "Buzz". If it is a multiple of 3 and 5, print "FizzBuzz". You can do this in any language, and there is no HackerRank solution for this.

2 Bitwise Manipulation

What does this function do (i.e. when does it return true/1?) There is no HackerRank solution for this problem, nor any code. Just a sentence describing when the function returns true is sufficient.

```
int func(unsigned int n){
    return n && !(n & (n - 1))
}
```

3   Linked Lists

How would you detect a *cycle* in a Linked List? A cycle is when the tail of
the Linked List points back somewhere into the list, rather than to Null.

You can write this in any language you'd like, although for HackerRank,
the problem will need to be in C++ or Java.

Example signature for C++:

```
/*
  Detect loop in a linked list
  List could be empty also
  Node is defined as
  struct Node
  {
     int data;
     struct Node *next;
  }
*/
int HasCycle(Node *head)
{
   // Complete this function
   // Do not write the main method
}
```

4 Binary Trees

Given the root of a tree, print the tree *inorder*. As a hint to those of you who took 367 a long time ago, the "in" part refers to where the given root is. There is also *pre*order and *post*order, which you can also implement for "Bonus points".

```
Example signature for C++:
Node is defined as

struct node
{
    int data;
    node *left;
    node *right;
};
void Inorder(node *root) {

}
```

5 Linked Lists

Merge two sorted Linked Lists into a single, sorted Linked List. Assume your function takes the heads of the two Linked Lists. HackerRank allows C++ and Java, but on paper you can do anything you wish.

```
Example signature for C++:
Node *MergeLists(Node *headA, Node *headB);
```

6 Linked Lists

Print a Linked List in reverse. You're given the head of the list you need
to print.

```
Example signature for C++:
void PrintInReverse(Node *head);
```

7 String Manipulation

You can use any language for this problem.
John has discovered various rocks. Each rock is composed of various elements, and each element is represented by a lower-case Latin letter from 'a' to 'z'. An element can be present multiple times in a rock. An element is called a gem-element if it occurs at least once in each of the rocks.

Given the list of N rocks with their compositions, display the number of gem-elements that exist in those rocks.

Example input:
abcdde
baccd
eeabg

Example output: 2 (a and b are the only elements in all of the rocks)

8 Binary Trees

Write a function that takes the root of a binary tree, and flips/mirrors the entire tree given. For example, a tree that has A as the root, B as the left child, and C as the right child, would become a tree with A as the root, C as the left child, and B as the right child. You must do this for the entire tree.

```
Example signature for C++:
Node is defined as

struct node
{
    int data;
    node *left;
    node *right;
};
node *mirror(node *root) {

}
```