Junior Knights: Introductory Recursion Homework

1. Write a recursive function that takes in a positive integer n and returns the sum of the positive integers upto n. (Thus, tri(10) should return 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55.) Please do NOT use the closed form formula for this problem. This is just an exercise to practice recursion. The function prototype is given below in Python, C++ and Java:

```
// Function Prototype in C++
int tri(int n) ;

// Function Prototype in Java
public static int tri(int n)

// Function Definition Line in Python
def tri(n):
```

2. The n^{th} Harmonic number is defined as the sum of the first n reciprocals of the positive integers. (For example, harmonic(3) equals 1 + ½ + 1/3.) Write a function harmonic that takes in a positive integer n and returns the nth harmonic number. The function prototype is given below in Python, C++ and Java:

```
// Function Prototype in C++
double harmonic(int n) ;

// Function Prototype in Java
public static double harmonic(int n)

// Function Definition Line in Python
def harmonic(n):
```

3. Write a recursive function that takes in a positive integer n and returns the product of its digits. For example, proddigits(452) would return 4*5*2 = 40. The function prototype is given below in Python, C++ and Java:

```
// Function Prototype in C++
int proddigits(int n) ;

// Function Prototype in Java
public static int proddigits(int n)

// Function Definition Line in Python
def proddigits (n):
```

4. You were shown the Towers of Hanoi solution in lecture, where a recursive function printed out each move necessary to solve the Towers problem. Now, consider writing a recursive function which calculates the total cost of solving the problem for a tower of n disks. For the purposes of this problem, define the cost of moving disk number k (from the top) as k. For example, cost(2) = 4, because you move the top disk (cost 1), followed by the disk below it initially (cost 2), followed by moving the top disk again (cost 1). Since the answer to this problem is the same no matter what the start and end towers are, the function will only take in a single positive integer, n, representing the number of disks to solve the problem for. The function prototype is given below in Python, C++ and Java:

```
// Function Prototype in C++
int towercost(int n) ;

// Function Prototype in Java
public static int towercost(int n)

// Function Definition Line in Python
def towercost(n):
```

5. You were shown the a recursive function that performs a binary search in lecture. That function returns true if the value searched for is in the sorted array, and false otherwise. Modify that code so that if the value is in the sorted array, it returns the index in which that value was found. If the value is not in the sorted, then -1 is returned.

```
// Function Prototype in C++
int binsearch(vector<int>& arr, int low, int high, int val) ;

// Function Prototype in Java
public static int binsearch(int[] arr, int low, int high, int
val);

// Function Definition Line in Python
def binsearch(arr,low,high,val):
```