ECSE207L
DATA STRUCTURES

Dr. Tapas Badal
Dept. of CSE
Bennett University

Tutorial 1:

a. Write a recursive function to calculate the sum of first n natural numbers.

```
int addNumbers(int n)
{
    if(n != 0)
        return n + addNumbers(n-1);
    else
        return n;
}
```

b. Write a recursive function to search an element in the given random array.

```
int recSearch(int arr[], int l,
              int r, int x)
{
   if (r < l)
      return -1;
   if (arr[l] == x)
      return l;
   return recSearch(arr, l + 1, r, x);
}
```

Write a recursive function to find the sum of the terms: 1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 +8^2.

```
//Sum of squares Given Series
// Recursive function to return
   // the sum of the given series
   static long sum(int n)
   {
       // 1^1 = 1
     if (n == 1)
         return 1;
      else
           // Recursive call
         return (n*n + sum(n - 1));
   }
```

```
    // Driver code
    public static void main(String 
args[])
    {
        int n = 4;
        System.out.println(sum(n));
    }
```

Make it a general function in terms of n and m to sum any series. Try your function by setting m = 3 and n = 8 to solve this particular series.

```
int summation(int n)
{
    int sum = 0;
    for (int i = 1; i <= n; i++)
        sum += (i * i *i);

    return sum;
}
```

Write a recursive function which returns the sum of elements of an array.

```
int findSum(int A[], int N)
{
    if (N < 0)
        return 0;
    return (findSum(A, N - 1) + A[N - 1]);
}
```

Write a recursive function which returns the sum of alternate elements of an array starting from the last element. Number of elements are even.

```
void EvenOddSum(int arr[], int n)
{
    int even = 0;
    int odd = 0;
    for (int i = 0; i < n; i++) {
        // Loop to find even, odd sum
        if (i % 2 == 0)
            even += arr[i];
        else
            odd += arr[i];
    }
}
```

Write a recursive function to print a string backwards.

```
class StringReverse
{
    /* Function to print reverse of the passed string */
    void reverse(String str)
    {
        if ((str==null)||(str.length() <= 1))
           System.out.println(str);
        else
        {
           System.out.print(str.charAt(str.length()-1));
           reverse(str.substring(0,str.length()-1));
        }
    }
    /* Driver program to test above function */
    public static void main(String[] args)
    {
        String str = "Bennett University";
        StringReverse obj = new StringReverse();
        obj.reverse(str);
    }
}
```

Write a recursive function to compute GCD of two integers a and b, given that a> b.

```
int gcd(int n1, int n2)
{
    if (n2 != 0)
       return gcd(n2, n1%n2);
    else
       return n1;
}
```

Write a recursive function to print the binary equivalent of a decimal integer.

```
findBinary(decimal)
   if (decimal == 0)
      binary = 0
   else
      binary = decimal % 2 + 10 * (findBinary(decimal / 2)
```

Write a recursive function to implement selection sort algorithm.

```
class SelectionSort
{
    // Utility function to swap values at two indices in the array
    public static void swap(int[] arr, int i, int j)
    {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
```

```
// Recursive function to perform selection sort on sub-array arr[i..n-1]
    public static void selectionSort(int[] arr, int i, int n)
    {
        // find the minimum element in the unsorted sub-array[i..n-1]
        // and swap it with arr[i]
        int min = i;
        for (int j = i + 1; j < n; j++)
        {
            // if arr[j] element is less, then it is the new minimum
            if (arr[j] < arr[min]) {
                min = j;    // update index of min element
            }
        }
        // swap the minimum element in sub-array[i..n-1] with arr[i]
        swap(arr, min, i);

        if (i + 1 < n) {
            selectionSort(arr, i + 1, n);
        }
    }
     public static void main(String[] args)
    {
        int arr[] = { 3, 5, 8, 4, 1, 9, -2 };

        selectionSort(arr, 0, arr.length);

        // print the sorted array
        System.out.println(Arrays.toString(arr));
    }
}
```

THANKYOU

@csebennett cse_bennett