

POWER SOLUTION

```
public int power(int base, int exponent) {  
    if (exponent == 0) {  
        return 1;  
    }  
    return base * power(base,exponent-1);  
}
```

FACTORIAL SOLUTION

```
public int factorial(int num) {  
    if (num<=1) {  
        return 1;  
    }  
    return num * factorial(num-1);  
}
```

PRODUCT OF ARRAY SOLUTION

```
public int productofArray(int A[], int N)  
{  
    if (N <= 0)  
        return 1;  
    return (productofArray(A, N - 1) * A[N - 1]);  
}
```

RECURSIVE RANGE SOLUTION

```
public int recursiveRange(int num) {  
    if (num<=0) {  
        return 0;  
    }  
    return num + recursiveRange(num - 1);  
}
```

FIBONACCI SOLUTION

```
public int fib(int n) {  
    if (n<0) {  
        return -1;  
    }  
    if (n==0 || n==1) {  
        return n;  
    }  
    return fib(n-1) + fib(n-2);  
}
```

REVERSE SOLUTION

```
public String reverse(String str)  
{  
    if (str.isEmpty())  
        return str;  
    //Calling Function Recursively  
    return reverse(str.substring(1)) + str.charAt(0);  
}
```

IS PALINDROME SOLUTION

```
public boolean isPalindrome(String s)  
{    // if length is 0 or 1 then String is palindrome  
    if(s.length() == 0 || s.length() == 1)  
        return true;  
    if(s.charAt(0) == s.charAt(s.length()-1))  
        return isPalindrome(s.substring(1, s.length()-1));  
    return false;  
}
```

SOME RECURSIVE SOLUTION

```
public boolean someRecursive(int[] arr, OddFunction odd) {
    if (arr.length == 0 ) {
        return false;
    } else if (odd.run(arr[0]) == false) {
        return someRecursive(Arrays.copyOfRange(arr, 1, arr.length), odd);
    } else {
        return true;
    }
}
```

FIRST UPPERCASE SOLUTION

```
static char first(String str)
{
    for (int i = 0; i < str.length(); i++)
        if (Character.isUpperCase(str.charAt(i)))
            return str.charAt(i);
    return 0;
}
```

CAPITALIZE WORD SOLUTION

```
public static String capitalizeWord(String str){
    String words[]=str.split("\\s");
    String capitalizeWord="";
    for(String w:words){
        String first=w.substring(0,1);
        String afterfirst=w.substring(1);
        capitalizeWord+=first.toUpperCase()+afterfirst+" ";
    }
    return capitalizeWord.trim();
}
```