5) The code below returns the number of zeros at the end of n! [factorial n]

```
int zeros(int n)
{
        int res = 0;
        while (n!=0)
        {
                res  += n/5;
                n /= 5;
        }
        return res;
}
```

```
int zeros (int n) {
    if (n==0);
        return 0;
    return zeros(n/5)+n/5;
}
```

Rewrite this method recursively:

6. Write a recursive function that returns the product of the digits of its integer input parameter, n. You omay assume that n is non-negative. For example, productDigits(243) should return 24, since 2 x 4 x 3 = 24.

```
int productDigits (int n) {
    if (n==0)
        return 1;
    return productDigits(n/10) * n%10;
}
```

7. Let us define the weighted sum of an integer array a[0], a[1], a[2], …, a[n-1] be a[0]*1 + a[1]*2 + a[2]*3 + …+a[n-1]*n. For example, the weighted sum of the array [5,2,6] would be 5*1+2*2+6*3 = 27. Write a recursive function that takes in an array numbers and its length n, and returns its weighter sum. You can assume n is non-negative integer.

```
int weightedSum(int numbers[], int n) {
    if (n==0)
        return 0;
    return numbers[n-1]*n + weightedSum(numbers, n-1);
}
```