8. Mathematically, given a function f, we recursively define $f^k(n)$ as follows: if $k = 1$, $f^1(n) = f(n)$. Otherwise, for $k > 1$, $f^k(n) = f(f^{k-1}(n))$. Assume that there is an existing function f, which takes in a single integer and returns an integer. Write a ***recursive*** function fcomp, which takes in both n and k (k > 0), and returns $f^k(n)$.

```
int f(int n);
int fcomp(int n, int k){
    if (k==1)
        return f(n);
    return f(fcomp(n, k-1));
}
```

9. What would be the value of fun(7) for the following function?

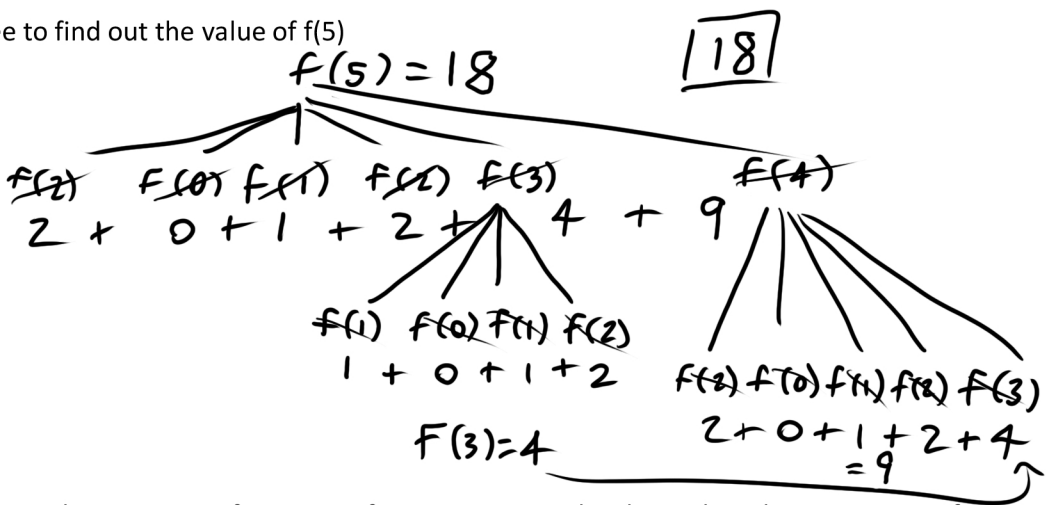
```
int fun(int x)
{
  if(x==0)
    return 0;
  if(x%3 ==0)
    return fun(x/3);
  return fun(x-1) + x;
}
```

fun(6)+7 = 10
fun(2) = 3
fun(1) +2 = 3
fun(0) + 1 = 1
0

10

10. Draw the recursion tree to find out the value of f(5)

```
int f(int n)
{
        int ans;
        int i;
        if(n<3)
                return n;
        ans = f(n/2);
        for(i=0; i<n; i++)
                ans += f(i);
        return ans;
}
```



11. Write a recursive function that returns 1 if an array of size n is in sorted order and 0 otherwise. Note: If array a stores 3, 6, 7, 7, 12, then isSorted(a, 5) should return 1 . If array b stores 3, 4, 9, 8, then isSorted(b, 4) should return 0.

```
int isSorted(int *array, int n){
    if (n<2)
      return 1;
    If (array[n-1]< array[n-2])
      return 0;
    return isSorted(array, n-1);
}
```