Practice with Big-O Notation

1) Let a and b be positive constants. Show that if f is $O(\log_a n)$ then f is also $O(\log_b n)$.

2) Suppose the size of the input is doubled for an algorithm (going from n to $2n$). Explain how the number of operations change for the algorithm if its Big-O complexity is
a) $O(n)$ b) $O(n^2)$ c) $O(\log(n))$ d) $O(2^n)$

What if the size of the input increases by 1 (going from n to $n+1$)?

3) Rank the following functions according to how fast they grow as $n \to \infty$: $n\log^2(n), \log^{2019}(n), \sqrt{n}, n^2, n!, \sqrt{2}^n, n^n$.

Determine the running time of the following program segments in Big-O notation. Take the size of the input as n, unless otherwise stated.

4)

```
double sum=0;
for(int i=0; i<1000000;i++)
    sum+=sqrt(i);
```

5)

```
while(n>1)
{
    n=n/2;
    cout<<"This is a useless code";
}
```

6)

```
int count=0;
for(i=0;i<n;i++)
{
    count++;
}
```

What happens if we take the size of the input as $\log(n)$ as opposed to n?

7)

```
for(i=0;i<n;i++)
{
    m=n;
    while(m>1)
    {
        m=m/2;
        cout<<m<<endl;
    }
}
```

8)

```
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        count++;

    }
}
```

a) Considering the size of the input as n
b) Considering the size of the input as $\log(n)$

9)

```
for(i=0;i<n;i++)
{
    for(j=i;j<n;j++)
    {
        count++;
    }
}
```

10)

```
int i, j,k;

for(k=0;k<n;k++)
{
 for(i=0;i<n;i++)
 {
    j=n;
    while(j>1)
    {
        j=j/3;
        cout<<i*j*k<<endl;
    }
 }
}

for(i=0;i<n;i++)
{
    cout<<i*i<<endl;
}
```