

# Program output from script file

---

Count of Perfect, Abundant, and Deficient numbers from 1 to 10000000  
Please wait. Running...

Count of Perfect Numbers = 4  
Count of Abundant Numbers = 2476737  
Count of Deficient Numbers = 7523258  
Total = 9999999

real 1m38.050s  
user 1m37.964s  
sys 0m0.024s

Count of Perfect, Abundant, and Deficient numbers from 1 to 10000000  
Please wait. Running...

Count of Perfect Numbers = 4  
Count of Abundant Numbers = 2476737  
Count of Deficient Numbers = 7523258  
Total = 9999999

real 1m36.708s  
user 1m36.645s  
sys 0m0.004s

Count of Perfect, Abundant, and Deficient numbers from 1 to 10000000  
Please wait. Running...

Count of Perfect Numbers = 4  
Count of Abundant Numbers = 2476737  
Count of Deficient Numbers = 7523258  
Total = 9999999

real 1m38.525s  
user 1m38.397s  
sys 0m0.048s

Count of Perfect, Abundant, and Deficient numbers from 1 to 10000000  
Please wait. Running...

Count of Perfect Numbers = 4  
Count of Abundant Numbers = 2476737  
Count of Deficient Numbers = 7523258  
Total = 9999999

real 0m33.226s  
user 1m39.010s

```
sys    0m0.244s
Count of Perfect, Abundant, and Deficient numbers from 1 to 10000000
Please wait. Running...
```

```
Count of Perfect Numbers = 4
Count of Abundant Numbers = 2476737
Count of Deficient Numbers = 7523258
Total = 9999999
```

```
real   0m33.405s
user   1m39.648s
sys    0m0.173s
Count of Perfect, Abundant, and Deficient numbers from 1 to 10000000
Please wait. Running...
```

```
Count of Perfect Numbers = 4
Count of Abundant Numbers = 2476737
Count of Deficient Numbers = 7523258
Total = 9999999
```

```
real   0m33.391s
user   1m39.669s
sys    0m0.120s
```

#### Explanation of the results (sequential and parallel) with calculated speedup

When the program is done sequentially, the processes are executed one after another, which takes longer than if the program is done in parallel, or when the program does not have to wait for the previous processes to be done. On average, the sequential option took about 1 minute and 37.761 seconds compared to the parallel option with only 33.341 seconds. The speedup would be  $\frac{37.761}{33.341} = 1.13$  times speedup.

#### Comment the lock and unlock calls and execute

After executing the commented lock and unlock calls, the sequential option took less time with 1 minute and 18.973 seconds with complete accuracy while the parallel option took more time with 1 minute and 48.177 seconds with incorrect results.

#### Copy of the bash script

```
#!/bin/bash

time ./numCount s 10000000
time ./numCount s 10000000
time ./numCount s 10000000

time ./numCount p 10000000
time ./numCount p 10000000
```

```
time ./numCount p 10000000
```