

## Parallel Sieve of Eratosthenes

```
#include <stdio.h>
#include <math.h>
#include <time.h>
#include "omp.h"
#define N 1000000
//Marked(value 1) means prime

char list[N+1];
void prime()
{
    int c,j,g=sqrt(N);
    #pragma omp parallel for private(j)
    for(c=2; c<=g; c++)
        if(list[c]==1)
            for(j=c*c;j<=N;j=j+c)
                list[j]=0;
}

int main()
{
    clock_t start, end;
    double cpu_time_used;
    start = clock();
    int count=0;
    unsigned long long i,c,j=0;
    omp_set_num_threads(3);
    // #pragma omp parallel for
    for(i=0; i<N; i++)
        list[i]=1;
    prime();
    for(i=0; i<N; i++)
    {
        if(list[i]==1)
        {
            //printf("%lld ",i);
            count++;
        }
    }
    end = clock();
    cpu_time_used = ((double) (end - start))/ CLOCKS_PER_SEC;
    printf("\nTime Taken: %f\n", cpu_time_used/3600);
    printf("\nNo of Primes: %d\n",count);
    return 0;
}
```

### Sequential Sieve of Eratosthenes

```
#include <stdio.h>
#include <math.h>
#include <unistd.h>
#include <time.h>
#define N 1000000

int main()
{
    clock_t start, end;
    double cpu_time_used;
    start = clock();
    unsigned long long i,j=0, list[N+1];
    int count=0;
    for(i=0; i<N; i++)
        list[i]=1;
    //Marked(value 1) means prime
    i=2;
    while(i<=sqrt(N))
    {
        while((list[i]!=1) && (i<=N))
            i++;
        j=i*i;
        while(j<=N)
        {
            list[j]=0;
            j+=i;
        }
        i++;
    }
    //Printing prime nos
    for(i=0; i<N; i++)
    {
        if(list[i]==1)
        {
            //printf("%lld ",i);
            count++;
        }
    }
    end = clock();
    cpu_time_used = ((double) (end - start))/ CLOCKS_PER_SEC;
    printf("\nTime Taken: %f\n", cpu_time_used/3600);
    printf("\nNo of Primes: %d\n",count);
    return 0;
}
```

```
7 998689 998717 998737 998743 998749 998759 998779 998813 998819 998831 998839 9
98843 998857 998861 998897 998909 998917 998927 998941 998947 998951 998957 9989
69 998983 998989 999007 999023 999029 999043 999049 999067 999083 999091 999101
999133 999149 999169 999181 999199 999217 999221 999233 999239 999269 999287 999
307 999329 999331 999359 999371 999377 999389 999431 999433 999437 999451 999491
999499 999521 999529 999541 999553 999563 999599 999611 999613 999623 999631 99
9653 999667 999671 999683 999721 999727 999749 999763 999769 999773 999809 99985
3 999863 999883 999907 999917 999931 999953 999959 999961 999979 999983
Time Taken: 0.000029

No of Primes: 78500
nishtha@nishtha-VPCEG3AEN:~/Downloads$ gcc -fopenmp sequential.c
nishtha@nishtha-VPCEG3AEN:~/Downloads$ ./a.out

Time Taken: 0.000013

No of Primes: 78500
nishtha@nishtha-VPCEG3AEN:~/Downloads$ gcc -fopenmp parallel.c
nishtha@nishtha-VPCEG3AEN:~/Downloads$ ./a.out

Time Taken: 0.000013

No of Primes: 78500
nishtha@nishtha-VPCEG3AEN:~/Downloads$ █
```