

```
1  /*
2  Author:  Pate Williams (c) 1997 - 2022
3  */
4  #pragma once
5  #include "Number_Theory.h"
6  #include <math.h>
7
8  struct factor { long expon; long long prime; };
9
10 int do_Brent_Method(
11     long long *N,
12     struct factor *f,
13     long *count);
14 /*
15 returns -1 failure
16 returns 0 complete factorization within factor base
17 returns 1 complete factorization with Miller-Rabin test
18 Algorithm 8.5.2 (Pollard rho). See "A Course in
19 Computational Algebraic Number Theory" by Henri
20 Cohen page 429. Brent's modification. The number count
21 is the number of factors.
22 */
23 long long first_stage(long long *n, int B1,
24     int *failure, long long *x, int number1,
25     long *primes1);
26 long long second_stage(
27     long long *n, int B1,
28     int B2, int *failure,
29     long numdiff, long *differs,
30     long number1, long *primes1,
31     long number2, long *primes2);
32 /*
33 returns -1 failure and failure = 1
34 returns g a factor otherwise
35 Algorithm 8.8.3 (p - 1 with Stage 2). See "A Course in
36 Computational Algebraic Number Theory" by Henri Cohen
37 page 441.
38 */
39 int trial_division(
40     long long *N,
41     struct factor *f,
42     long *count,
43     long psize,
44     long *prime);
45 /*
46 returns 1 for success and 0 for failure
47 Algorithm 8.1.1 (Trial Division). See "A Course in
48 Computational Algebraic Number Theory" by Henri
49 Cohen page 420. The number count is the number of
50 factors and psize is the number of primes.
51 */
52 void QuickSort(struct factor *f, int n);
```

```
53 /* Quick sort algorithm from "Algoritihms" by
54    Thomas H. Cormen, et. al. */
```