

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <pthread.h>
5
6  #define THREAD_FUNC(f) (void*)(*)(void*))f
7
8  typedef struct
9  {
10     int x;
11     int verrou;
12 } data;
13
14
15 /* La section critique est formée d'une seule ligne :
16  * --(*px);
17  * car c'est le seul moment où on manipule une donnée qui est commune aux deux
18  * threads.
19  */
20 int * decomppte( data * pd )
21 {
22     int y;
23     int *py;
24     while ( pd->x > 0 )
25     {
26         ++y;
27         while ( pd->verrou == 1 )
28             { /* ne rien faire */ }
29         pd->verrou = 1;
30         --(pd->x);
31         pd->verrou = 0;
32     }
33     py = ( int* ) malloc( sizeof( int ) );
34     *py = y;
35     pthread_exit( py );
36 }
37
38 int main( int argc, const char ** argv )
39 {
40     data d;
41     int *ya, *yb;
42     pthread_t ta, tb;
43     (void) argc; /* ne fait rien, évite un warning si compilé avec -Wextra */
44     d.x = atoi( argv[1] );
45     d.verrou = 0;
46     pthread_create( &ta, NULL, THREAD_FUNC( decomppte ), (void*) &d );
47     pthread_create( &tb, NULL, THREAD_FUNC( decomppte ), (void*) &d );
48     pthread_join( ta, (void**) &ya );
49     pthread_join( tb, (void**) &yb );
50     printf( "%d + %d = %d\n", *ya, *yb, *ya + *yb );
51     free( ya );
52     free( yb );
53     return 0;
54 }

```