```
#include <pthread.h>
#include <stdio.h>

int sum;
void *runner(void *parm);

main(int argc, char *argv[])
{
        pthread_t tid;
        pthread_attr_t attr;

        pthread_attr_init(&attr);
        pthread_create(&tid,&attr,runner,argv[1]);
        pthread_join(tid,NULL);
        printf("sum = %d\n",sum);
}

void *runner(void *parm)
{
        int i;
        sum = 0;
        for (i = 1; i <= atoi(parm); i++)
                sum += i;
        pthread_exit(0);
}

$ gcc example.c -o example -lpthread
$ ./example 20
sum = 210
```

1 Include the Pthreads library; all Pthread programs must include `pthread.h`

5 When the program begins, a single thread of control begins in `main`. After some initialization, `main` creates a second thread that begins control in the `runner` function.

7 Declare the identifier for the thread we will create.

8 Each thread has a set of attributes including stack size and scheduling information.

10 Set the attributes; because we did not explicitly set any attributes, we use the default attributes provided.

11 A separate thread is created: we pass the thread identifier, the attributes for the thread, the function name where the thread will start execution (`runner` function), and the integer parameter provided on the command line (`argv[1]`).

12 At this point, the program has two threads: the initial thread in `main` and the thread performing the summation in the runner function. After creating the second thread, the main thread will wait for the runner thread to complete by calling the pthread_join function. The runner thread will complete when it calls the function pthread_exit.