```c
/***********************************************
 * collatz: get a number from user input       *
 *          and check Collatz conjecture       *
 *          code example 2 from lecture 3      *
 *          dated 24.04.19                     *
 ***********************************************/

#include <stdio.h>

/*
    The macro ITERATION_MAX restricts the number of iterations and
    helps to avoid an infinite loop in the case that the collatz
    conjecture turns out to be wrong.
*/
#define ITERATION_MAX 500

int main(){
    /*
        Use size_t for large positive numbers when integer overflow
        could occur for large int's.
    */
    size_t n;
    printf("Please enter a number: ");
    /*
        Interpret user input as a number, convert it into the size_t
        data type and write the result to memory address &n. We always
        need to remember that C only passes the values of the variables
        to functions, not the variables themselves. Thus, when n is passed,
        a copy of the variable with the same value would be created,
        and the original variable would remain unmodified throughout the action
        of the function. This evaluation strategy is referred to as
        'call by value'. If we want, that a function changes the value of
        a variable, we need to pass the memory address of said variable.
        Here we do this utilising the address operator '&'.
    */
    scanf("%zu",&n);

    printf("Checking conjecture for %zu...\n",n);

    // First define variable, that keeps track of iteration number.
    size_t iterations = 0;

    // Iterate while n is unequal one
    while (n != 1){
        if (n % 2 == 0){
            n = n / 2;
        } else {
            n = 3*n + 1;
        }

        // don't forget to increase the iteration counter
        iterations = iterations + 1;

        // avoid infinite loops, stop after ITERATION_MAX is reached.
        if (iterations >= ITERATION_MAX){
            break;
        }
    }

    // print the result
    if(n == 1){
        printf("Conjecture correct for this number.\n");
        printf("Iterations required: %zu\n",iterations);
    } else {
        printf("Iteration limit %zu reached, aborting.\n",iterations);
    }

    return 0;
}
```