

Assignment 5

Introduction to programming in C

Question 1

The Collatz function is defined for a positive integer n as follows.

$$f(n) = \begin{cases} 3n + 1, & \text{if } n \text{ odd} \\ n/2, & \text{if } n \text{ is even} \end{cases}$$

We consider the repeated application of the Collatz function starting with a given integer n , as follows: $f(n), f(f(n)), f(f(f(n))), \dots$

It is conjectured that no matter which positive integer n you start from, this sequence eventually will have 1 in it. It has been verified to hold for numbers up to 5×260 .

For example, if $n=7$, the sequence is $f(7) = 22, f(f(7)) = f(22) = 11, f(11) = 34, f(34) = 17, f(17) = 52, f(52) = 26, f(26) = 13, f(13) = 40, f(40) = 20, f(20) = 10, f(10) = 5, f(5) = 16, f(16) = 8, f(8) = 4, f(4) = 2, f(2) = 1$.

Thus if you start from $n=7$, you need to apply f 16 times in order to first get 1.

In this question, you will be given a positive number $\leq 32,000$. You have to output how many times f has to be applied repeatedly in order to first reach 1.

Solution

```
1 #include <stdio.h>
2
3 int collatz_repeat(int n) {
4     if (n == 1) {
5         return 0;
6     } else {
7         if (n % 2 == 1) {
8
9             return 1 + collatz_repeat(3 * n + 1);
10        } else {
11            return 1 + collatz_repeat(n / 2);
12        }
13    }
14 }
15
```

```

16 int main() {
17     int n;
18     scanf("%d", &n);
19     printf("%d", collatz_repeat(n));
20     return 0;
21 }

```

Question 2

Given an integer array M having size n which is power of 2, Write a recursive code to find the BlockSum of the array M.

The following is the recursive definition of BlockSum:

If size of M is 2, say $M = [a, b]$, where a and b are integers, then

$$BlockSum(M) = a - b$$

Otherwise (when $n > 2$), partition M into two subarrays of equal size:

$$M = [A \quad B \quad C \quad D]$$

The BlockSum of M is defined recursively as :

$$BlockSum(M) = BlockSum(A) - BlockSum(B).$$

Here A and B are arrays of Size n/2 each. A is the first n/2 elements of M (in the same order) and B is the last n/2 elements of M (in the same order).

Note : You can assume that size of input array is a power of 2, and the size is less than 1024.

Input

The first line contains the array size n.

The next n lines contains the elements of the array.

Output

BlockSum(M)

Solution

```
1 #include <stdio.h>
2
3 #define N 1024
4
5 int BlockSum(int M[N], int i, int j, int n)
6 {
7     int A,B;
8
9     if(n == 2){
10         return M[i] - M[j];
11     }
12
13     A = BlockSum(M, i, i+(n/2)-1, n/2);
14     B = BlockSum(M, i+(n/2), j, n/2);
15
16     return A - B;
17 }
18
19
20 int main()
21 {
22
23     int n;
24     int M[N];
25
26     scanf("%d",&n);
27
28     for(int i=0; i<n; i++)
29         scanf("%d", &M[i]);
30
31     printf("%d", BlockSum(M, 0, n-1, n));
32     return 0;
33 }
```