

## Conjetura de Collatz

La conjetura de Collatz, conocida también como conjetura  $3n+1$  o conjetura de Ulam (entre otros nombres), fue enunciada por el matemático Lothar Collatz en 1937, y a la fecha no se ha resuelto.

Sea la siguiente operación, aplicable a cualquier número entero positivo:

- Si el número es par, se divide entre 2.
- Si el número es impar, se multiplica por 3 y se suma 1.

$$f(n) = \begin{cases} \frac{n}{2}, & \text{si } n \text{ es par} \\ 3n + 1, & \text{si } n \text{ es impar} \end{cases}$$

Dado un número cualquiera, podemos considerar su órbita, es decir, las imágenes sucesivas al iterar la función. Por ejemplo, si  $n=13$ :

$$f(13) = 13 \cdot 3 + 1 = 40$$

$$f(f(13)) = \frac{40}{2} = 20$$

$$f(f(f(13))) = \frac{20}{2} = 10; \text{ etc}$$

Si observamos este ejemplo, la órbita de 13 es periódica, es decir, se repite indefinidamente a partir de un momento dado):

13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, ...

La conjetura dice que siempre alcanzaremos el 1 (y por tanto el ciclo 4, 2, 1) para cualquier número con el que comencemos. Ejemplos:

- Comenzando en  $n = 6$ , uno llega a la siguiente sucesión:

6, 3, 10, 5, 16, 8, 4, 2, 1.

- Empezando en  $n = 11$ , la sucesión tarda un poco más en alcanzar el 1:

11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Se desea implementar en lenguaje C un programa con una función **recursiva**  $f(n)$  que lea en el archivo de entrada `input.txt` (ver Figura 1) y produzca el archivo `output.txt` (ver Figura 1).

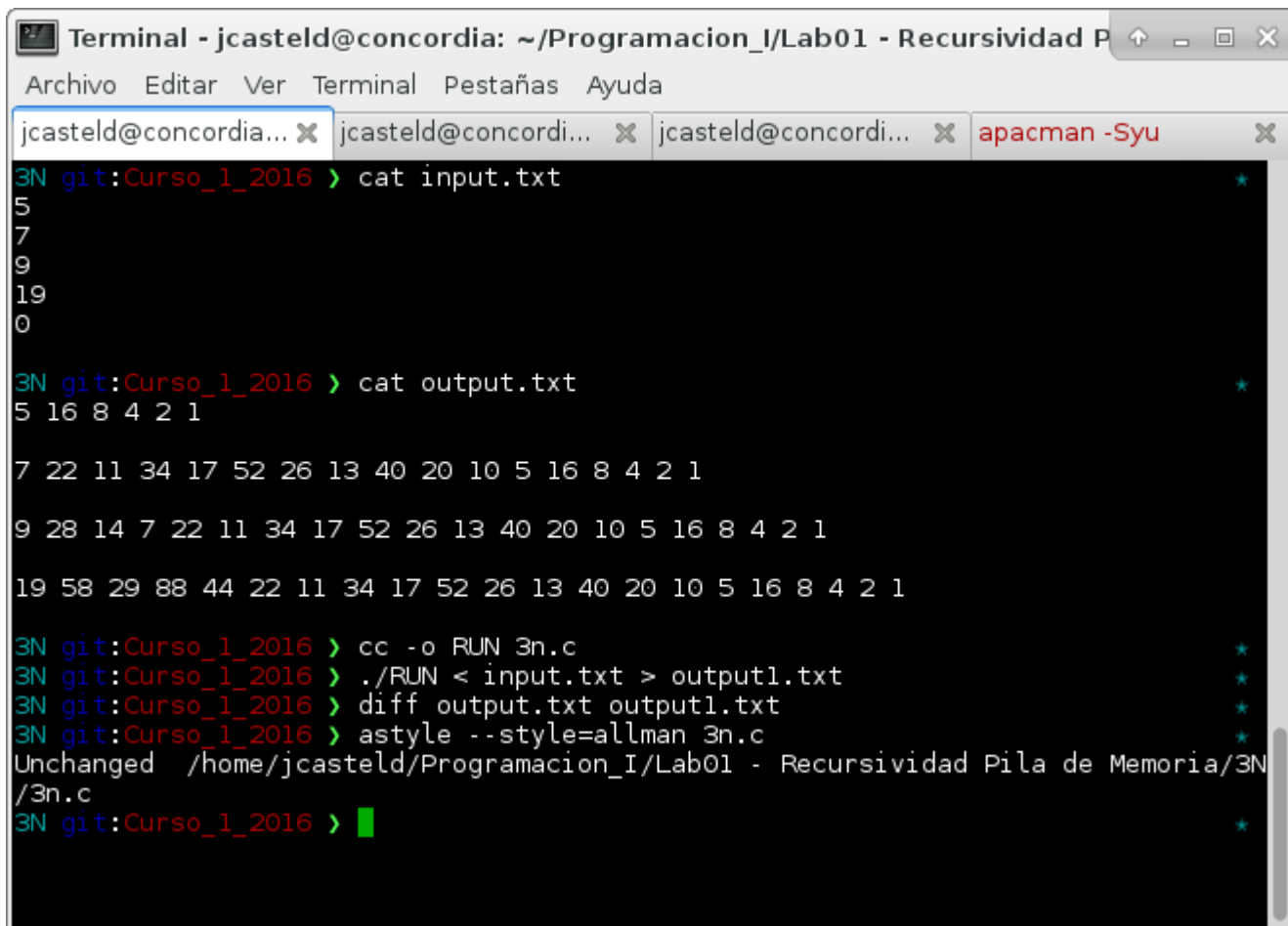
Observe que el archivo `input.txt` contiene un conjunto de valores de  $n$  y en el archivo `output.txt` se escriben todas las imágenes sucesivas al iterar la función  $f$  hasta finalizar con el valor 1.

En la Figura 2 se muestra el código del programa `3n.c` para que lo tipee, lo ejecute y sirva de modelo para ejercicios posteriores. Este paso lo puede sustituir con una versión propia que logre el mismo objetivo.

Una vez tipeado el código fuente `3n.c`, según se observa en la Figura 2 proceda a:

1. Crear el archivo `input.txt`.
2. Crear el archivo `output.txt`
3. Compilar el programa fuente `3n.c` y crear el archivo ejecutable `RUN`
4. Ejecutar el programa.

5. Compare la salida del archivo generado por el programa (output1.txt) con el archivo de resultados (output.txt) que creó previamente.
6. Formatee el archivo fuente con el comando `astyle` para que luzca apropiadamente.



```
Terminal - jcasteld@concordia: ~/Programacion_I/Lab01 - Recursividad P
Archivo Editar Ver Terminal Pestañas Ayuda
jcasteld@concordia... x jcasteld@concordi... x jcasteld@concordi... x apacman -Syu x
3N git:Curso_1_2016 > cat input.txt
5
7
9
19
0

3N git:Curso_1_2016 > cat output.txt
5 16 8 4 2 1

7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

19 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

3N git:Curso_1_2016 > cc -o RUN 3n.c
3N git:Curso_1_2016 > ./RUN < input.txt > output1.txt
3N git:Curso_1_2016 > diff output.txt output1.txt
3N git:Curso_1_2016 > astyle --style=allman 3n.c
Unchanged /home/jcasteld/Programacion_I/Lab01 - Recursividad Pila de Memoria/3N
/3n.c
3N git:Curso_1_2016 > █
```

Figura 1



```
1  #include <stdio.h>
2  /* 3n + 1 Version Recursiva
3     Version original por Kiara Ottogalli
4     Modificada por Jorge Castellanos
5  */
6
7  void f(int n);
8
9  int main()
10 {
11     int n;
12
13     scanf("%d", &n);
14     while(n > 0)
15     {
16         f(n);
17         printf("\n");
18         scanf("%d", &n);
19     }
20     return 0;
21 }
22
23 void f(int n)
24 {
25     if(n == 1)
26         printf("%d\n", n);
27     else
28     {
29         printf("%d ", n);
30         if(n % 2) /* n % 2 != 0 - impar */
31             f(3 * n + 1);
32         else /* n % 2 == 0 - par */
33             f(n >> 1); /* n / 2 */
34     }
35 }
```

Figura 2