Proyecto 2A: Crecimiento de la Población

Objetivo:

Determine cuánto tiempo le toma a una población alcanzar una determinada cantidad.

Contexto:

Supongamos que tenemos una población de llamas.

Al final de cada año han nacido n / 3 llamas, y han muerto n / 4 llamas.

Así, si tuviésemos 1200 llamas, al final del año, nacerían 1200 / 3 = 400 llamas y morirían 1200 / 4 = 300 llamas. Por lo que resultarían 1200 + 400 - 300 = 1300 llamas.

Si fuesen 1000 llamas, al final del año, nacerían 1000 / 3 = 333,33 llamas. Pero la cantidad de llamas debe ser un número natural, así que truncamos el decimal para obtener 333 llamas. Morirían 1000 / 4 = 250 llamas, por lo que resultan 1000 + 333 - 250 = 1083 llamas.

Empezando:

Puedes hacer clic en el siguiente enlace https://github.com/Davatec/Informat donde podrás descargar la carpeta que contiene el archivo **crecimiento.c** y las librerías cs50.c y cs50.h para que puedas realizar este proyecto.

Detalles:

Completa la implementación de **crecimiento.c**, de modo que calcule la cantidad de años necesarios para que la población crezca desde la cantidad inicial hasta la cantidad final.

Tu programa primero debe solicitar al usuario una cantidad de población inicial. Si el usuario ingresa un número menor que 9, se le debe solicitar de nuevo hasta que ingrese un número mayor o igual a 9. La razón de esto es que, si comenzamos con menos de 9 llamas, esta población se estancaría rápidamente.

Luego, tu programa debe solicitar al usuario una cantidad de población final. Si el usuario ingresa un número menor que la cantidad de población inicial, se le debe solicitar de nuevo que ingrese una cantidad hasta que ingrese un número mayor o igual que la cantidad de población inicial. La razón de esto, es que queremos que la población de llamas crezca.

Sugerencias:

Si desea volver a solicitar repetidamente al usuario el valor de una variable hasta que se cumpla alguna condición, puede utilizar un ciclo **do while**. Por ejemplo, el siguiente código solicita al usuario repetidamente hasta que ingrese un número natural.

```
int n;
do
{
    n = get_int("Natural: ");
} while (n < 1);
```

Puedes adaptar este código para asegurar una cantidad inicial de al menos 9 y una cantidad final de al menos la cantidad inicial.

Para declarar una nueva variable, por ejemplo, para realizar un seguimiento de cuántos años han pasado, asegúrate de especificar su tipo de datos, un nombre para la variable y (opcionalmente) cuál debería ser su valor inicial.

Para calcular cuántos años le tomará a la población alcanzar el tamaño final, podría ser útil usar otro bucle, dentro del cual podría actualizar el tamaño de la población de acuerdo con la fórmula indicada en la introducción y actualizar la cantidad de años que han pasado.

Para imprimir un entero n en la terminal, puedes usar una línea de código como

```
printf("El valor es: %i\n", n);
```

para especificar que la variable n debe rellenar el marcador de posición %i.

Prueba tu Código:

Su programa debe comportarse según estos ejemplos a continuación.

> ./crecimiento

Cantidad inicial: 100

Cantidad final: 200

Ciclos anuales: 9

> ./crecimiento

Cantidad inicial: 1200

Cantidad final: 1300

Ciclos anuales: 1

> ./ crecimiento

Cantidad inicial: -5

Cantidad inicial: 3

Cantidad inicial: 9

Cantidad final: 5

Cantidad final: 18

Ciclos anuales: 8

> ./ crecimiento

Cantidad inicial: 20

Cantidad final: 1

Cantidad final: 10

Cantidad final: 100

Ciclos anuales: 20

> ./ crecimiento

Cantidad inicial: 100

Cantidad final: 1000000

Ciclos anuales: 115