

# Laboratório 1: crescimento populacional

Determine quanto tempo leva para uma população atingir um determinado tamanho.

```
$ ./population  
Start size: 100  
End size: 200  
Years: 9
```

## Background

Digamos que temos uma população de  $n$  lhamas. A cada ano, nascem  $n / 3$  novas lhamas e  $n / 4$  morrem.

Por exemplo, se começarmos com  $n = 1.200$  lhamas, no primeiro ano,  $1.200 / 3 = 400$  novas lhamas nascerão e  $1.200 / 4 = 300$  lhamas morrerão. No final daquele ano, teríamos  $1.200 + 400 - 300 = 1.300$  lhamas.

Para tentar outro exemplo, se começarmos com  $n = 1000$  lhamas, no final do ano teremos  $1000/3 = 333,33$  novas lhamas. Não podemos ter uma parte decimal de uma lhama, entretanto, vamos truncar o decimal para que  $333$  novas lhamas nasçam.  $1000/4 = 250$  lhamas passarão, então terminaremos com um total de  $1000 + 333 - 250 = 1083$  lhamas no final do ano.

### CC50: Lab 1 - Crescimento Populacional



## Começando

Copie o “código de distribuição” (ou seja, código inicial) a seguir em um novo arquivo em seu IDE chamado population.c .

```
#include  
#include  
  
int main(void)  
{  
    // TODO: Solicite o valor inicial ao usuário  
  
    // TODO: Solicite o valor final ao usuário  
  
    // TODO: Calcule o número de anos até o limite  
  
    // TODO: Imprima o número de anos
```

```
}
```

## Detalhes de Implementação

Conclua a implementação de `population.c`, de forma que calcule o número de anos necessários para que a população cresça do tamanho inicial ao tamanho final.

Seu programa deve primeiro solicitar ao usuário um tamanho inicial da população.

Se o usuário inserir um número menor que 9 (o tamanho mínimo permitido da população), o usuário deve ser solicitado novamente a inserir um tamanho inicial da população até inserir um número maior ou igual a 9. (Se começarmos com menos de 9 lhamas, a população de lhamas ficará estagnada rapidamente!)

Seu programa deve então solicitar ao usuário o tamanho final da população.

Se o usuário inserir um número menor que o tamanho da população inicial, ele deverá ser solicitado novamente a inserir um tamanho da população final até inserir um número que seja maior ou igual ao tamanho da população inicial. (Afinal, queremos que a população de lhamas cresça!)

Seu programa deve então calcular o número (inteiro) de anos necessários para que a população atinja pelo menos o tamanho do valor final.

Finalmente, seu programa deve imprimir o número de anos necessários para que a população de lhama alcance esse tamanho final, como ao imprimir no terminal **Years: n**, onde **n** é o número de anos.

## Dicas

Se você deseja solicitar repetidamente ao usuário o valor de uma variável até que alguma condição seja atendida, você pode usar um loop do ... while. Por exemplo, recupere o seguinte código da palestra, que avisa o usuário repetidamente até que ele insira um número inteiro positivo.

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

Como você pode adaptar este código para garantir um tamanho inicial de pelo menos 9, além de um tamanho final que seja pelo menos o tamanho inicial?

Para declarar uma nova variável, certifique-se de especificar seu tipo de dado, um nome para a variável e (opcionalmente) qual deve ser seu valor inicial.

Por exemplo, você pode querer criar uma variável para controlar quantos anos se passaram.

Para calcular quantos anos a população levará para atingir o tamanho final, outro ciclo pode ser útil! Dentro do loop, você provavelmente desejará atualizar o tamanho da população de acordo com a fórmula em `Background` e atualizar o número de anos que se passaram.

Para imprimir um inteiro `n` no terminal, lembre-se de que você pode usar uma linha de código como

```
printf("O número é %i\n", n);
```

para especificar que a variável `n` deve ser preenchida para o espaço reservado `%i`.

## Como testar seu código

Seu código deve resolver os seguintes casos de teste:

- \$ ./population  
Start size: 1200  
End size: 1300  
Years: 1

- \$ ./population  
Start size: -5  
Start size: 3  
Start size: 9  
End size: 5  
End size: 18  
Years: 8

- \$ ./population  
Start size: 20  
End size: 1  
End size: 10  
End size: 100  
Years: 20

- \$ ./population  
Start size: 100  
End size: 1000000  
Years: 115

Execute o seguinte comando para avaliar a exatidão do seu código usando **check50**. Mas certifique-se de compilar e testar você mesmo!

```
check50 cs50/labs/2021/x/population
```

Execute o seguinte comando para avaliar o estilo do seu código usando **style50**.

```
style50 population.c
```

Em caso de dúvida, envie email para  
[relacionamento@estudar.org.br](mailto:relacionamento@estudar.org.br)