

This is CS50.

Think.

Pair.

Share.

cs50.ly/questions

- Why are we using **C**?
- How can we **read** and **write** code that includes **variables**, **conditionals**, and **loops**?
- Why do we care about **data types**?
- What does it mean to **compile** a C program?
- How many years will it take to double our llamas?

SECOND EDITION

THE

C



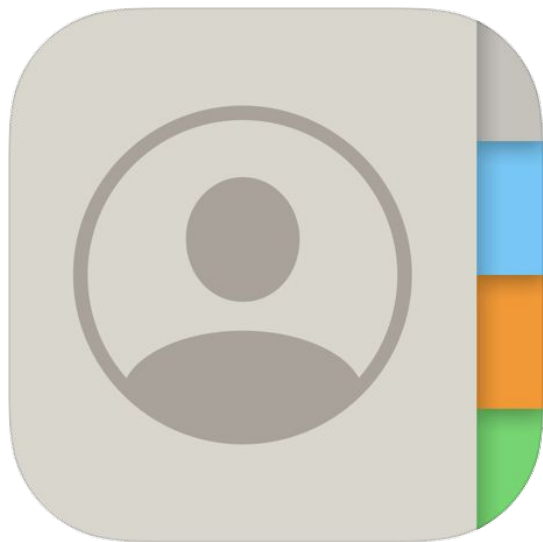
PROGRAMMING
LANGUAGE

BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

PRENTICE HALL SOFTWARE SERIES

Part 1

Variables and Types
Input and Printing



Variables

| |
|-------|
| Calls |
| 4 |

Variables

```
int calls = 4;
```

| |
|-------|
| Calls |
| 4 |

Variables

```
int calls = 4;
```

name

| |
|-------|
| Calls |
| 4 |

Variables

```
int calls = 4;
```

type

| |
|-------|
| Calls |
| 4 |

Variables

```
int calls = 4;
```

value

| |
|-------|
| Calls |
| 4 |

Variables

```
int calls = 4;
```



assignment
operator

| |
|-------|
| Calls |
| 4 |

Variables

int calls = 4;

type name | value

assignment
operator

| |
|-------|
| Calls |
| 4 |

"Create an **integer** named **calls** that **gets** the **value 4**."

Variables

```
int x = 50;
```

| |
|----|
| x |
| 50 |

Variables

```
int x = 50;
```

| |
|----|
| x |
| 50 |

"Create an **integer** named **x** that **gets** the **value 50**."

Think.
Pair.
Share.

Why does C care
about data types?

01000001

int

65

01000001

char

'A'

01000001

Variables

```
int calls = 4;  
calls = 5;
```

| Calls |
|-------|
| 5 |

Variables

```
int calls = 4;
```

```
calls = 5;
```

name

|

value

assignment
operator

| |
|-------|
| Calls |
| 5 |

"Calls gets 5."

Operators

```
int calls = 4;  
calls = calls + 1;
```

| Calls |
|-------|
| 5 |

Operators

```
int calls = 4;  
calls = calls - 1;
```

| Calls |
|-------|
| 3 |

Operators

```
int calls = 4;  
calls = calls * 2;
```

| Calls |
|-------|
| 8 |

Operators

```
int calls = 4;  
calls = calls / 2;
```

| Calls |
|-------|
| 2 |

Getting input

```
int calls = get_int("Calls: ");
```

type

name

|

function

assignment
operator

Functions

```
int calls = get_int("Calls: ");
```

function

Functions

```
int calls = get_int("Calls: ");
```

function name

Functions

```
int calls = get_int("Calls: ");
```

function input

Functions

```
int calls = get_int("Calls: ");
```

function

Return values

```
int calls = 4;
```

value

Storing return values

int calls = 4;

type name | value

assignment
operator

| |
|-------|
| Calls |
| 4 |

"Create an **integer** named **calls** that **gets** the **value 4**."

Printing values

```
int calls = 4;  
printf("calls equals %i", calls);
```

Printing values

```
int calls = 4;  
printf("calls equals %i", calls);
```



format code

Printing values

```
int calls = 4;  
printf("calls equals %i", calls);
```

Diagram illustrating the components of the `printf` function call:

- `%i` is labeled as the **placeholder** (indicated by a green underline).
- `calls` is labeled as the **value** (indicated by a purple underline).

Types and format codes

Numbers

Text

True/False

`int (%i)`

`char (%c)`

`bool (%i)`

`float (%f)`

`string (%s)`

Exercise

Create a C program that prompts a user for:

- A name
- An age
- A phone number

Print the values back to the user as confirmation.

Part 2

Breaking down loops
and conditionals

```
if (calls < 1)
{
    printf("Call more often!");
}
```

boolean expression



```
if (calls < 1)
{
    printf("Call more often!");
}
```


conditional



```
if (calls < 1)
{
    printf("Call more often!");
}
```

```
if (calls < 1)
{
    printf("Call more often!");
}
```



conditional code

```
if (calls < 1)
{
    printf("Call more often!");
}
else
{
    printf("Thanks for calling!");
}
```

```
if (calls < 1)
{
    printf("Call more often!");
}
else
{
    printf("Thanks for calling!");
}
```

↑
mutually exclusive
↓

```
int i = 0;
while (i < 10)
{
    printf("%i\n", i);
    i = i + 1;
}
```

initialization



```
int i = 0;  
while (i < 10)  
{  
    printf("%i\n", i);  
    i = i + 1;  
}
```

boolean expression

```
int i = 0;  ↓  
while (i < 10)  
{  
    printf("%i\n", i);  
    i = i + 1;  
}
```

```
int i = 0;  
while (i < 10)  
{  
    printf("%i\n", i);  
    i = i + 1;  
}
```



increment


```
int i = 0;  
while (i < 10)  
{  
    printf("%i\n", i);  
    i = i + 1;  
}
```

```
for (int i = 0; i < 10; i++)  
{  
    printf("%i\n", i);  
}
```

initialization



```
for (int i = 0; i < 10; i++)  
{  
    printf("%i\n", i);  
}
```

boolean expression



```
for (int i = 0; i < 10; i++)  
{  
    printf("%i\n", i);  
}
```

increment



```
for (int i = 0; i < 10; i++)  
{  
    printf("%i\n", i);  
}
```

```
for (int i = 0; i < 10; i++)  
{  
    printf("%i\n", i);  
}
```

```
int n;  
do  
{  
    n = get_int("N: ");  
}  
while (n <= 0);
```

```
int n;  
do  
{  
    n = get_int("N: ");  
}  
while (n <= 0);
```



```
int n;  
do  
{  
    n = get_int("N: ");  
}  
while (n <= 0);
```

Part 3

Population

- Work an example yourself
- Write down exactly what you did
- Create a generalization (algorithm) after working multiple examples
- Test your algorithm by hand
- Translate your algorithm to code
- Find bugs in your code by running test cases
- Debug (and critique) your code

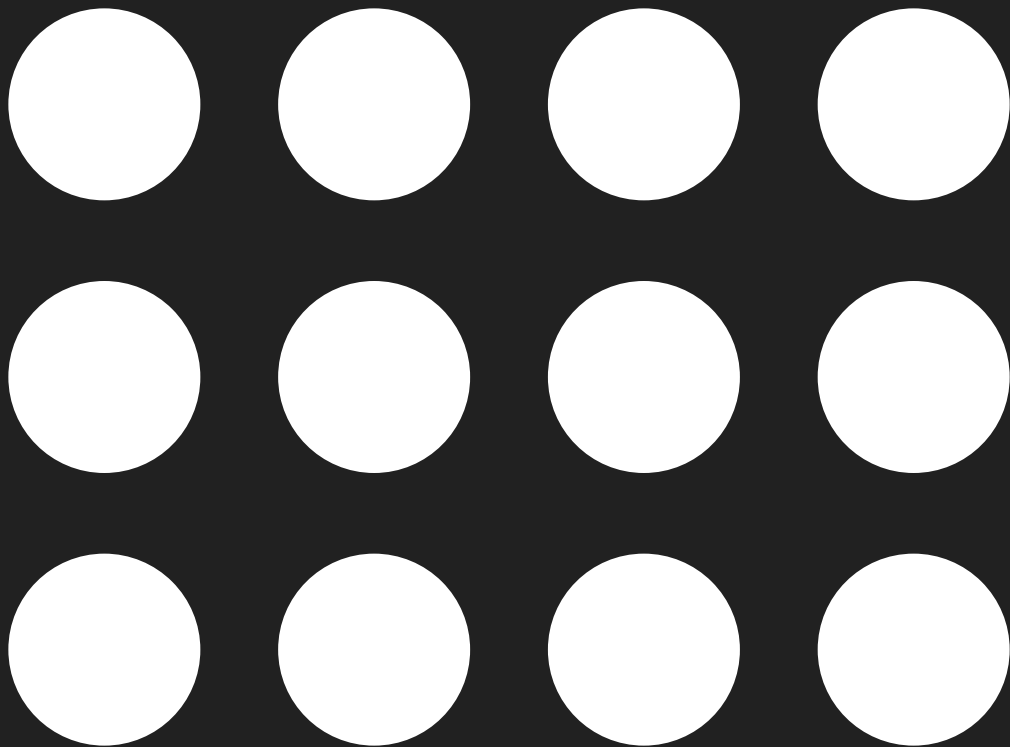
- Work an example yourself
- Write down exactly what you did
- Create a generalization (algorithm) after working multiple examples
- Test your algorithm by hand
- Translate your algorithm to code
- Find bugs in your code by running test cases
- Debug (and critique) your code

We have a population of n llamas.

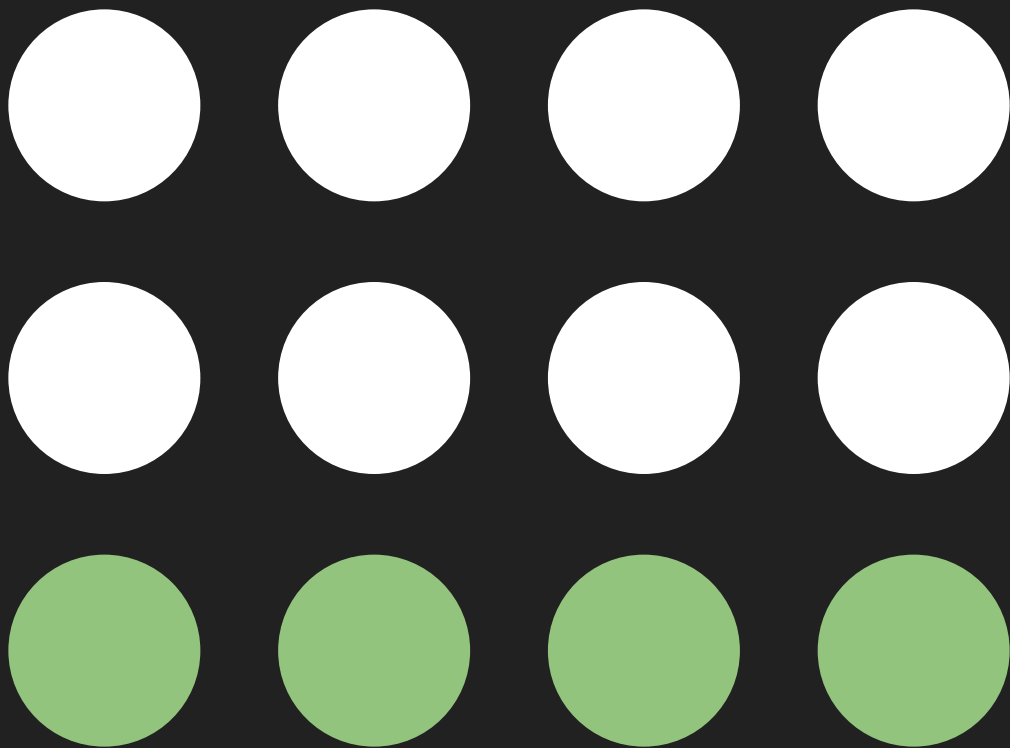
Each year, $n/3$ new llamas are born,
and $n/4$ llamas pass away.

How many years will it take to have a
population of x llamas?

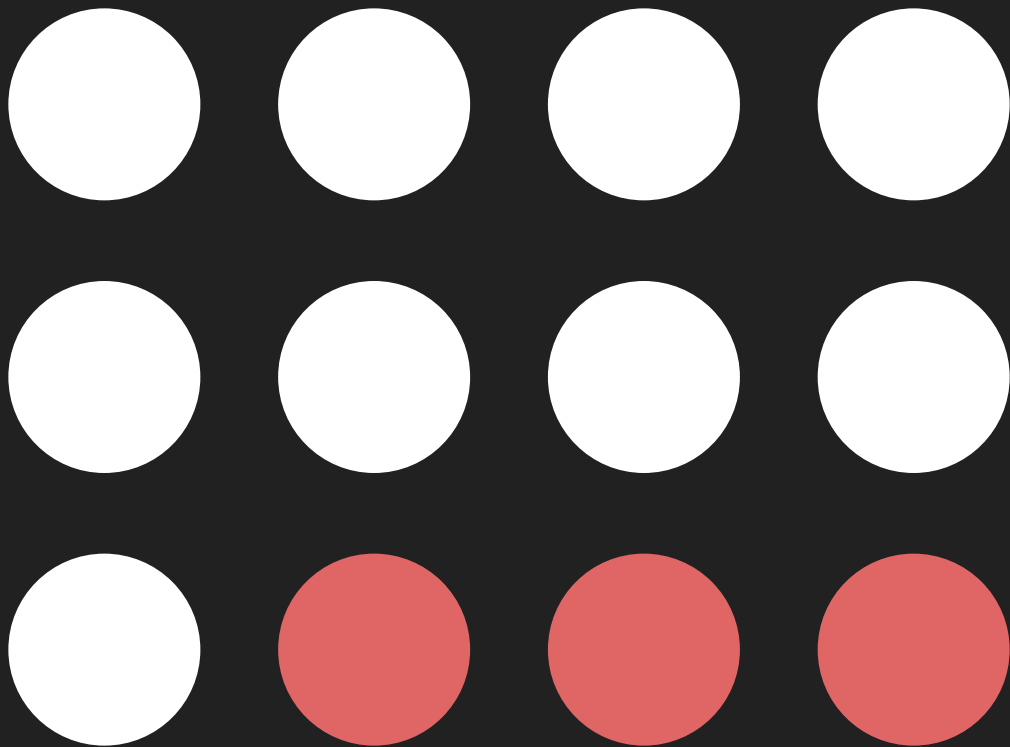
We have a population of **12** llamas.
Each year, **$12/3$** new llamas are born,
and **$12/4$** llamas pass away.
How many years will it take to have a
population of **13** llamas?



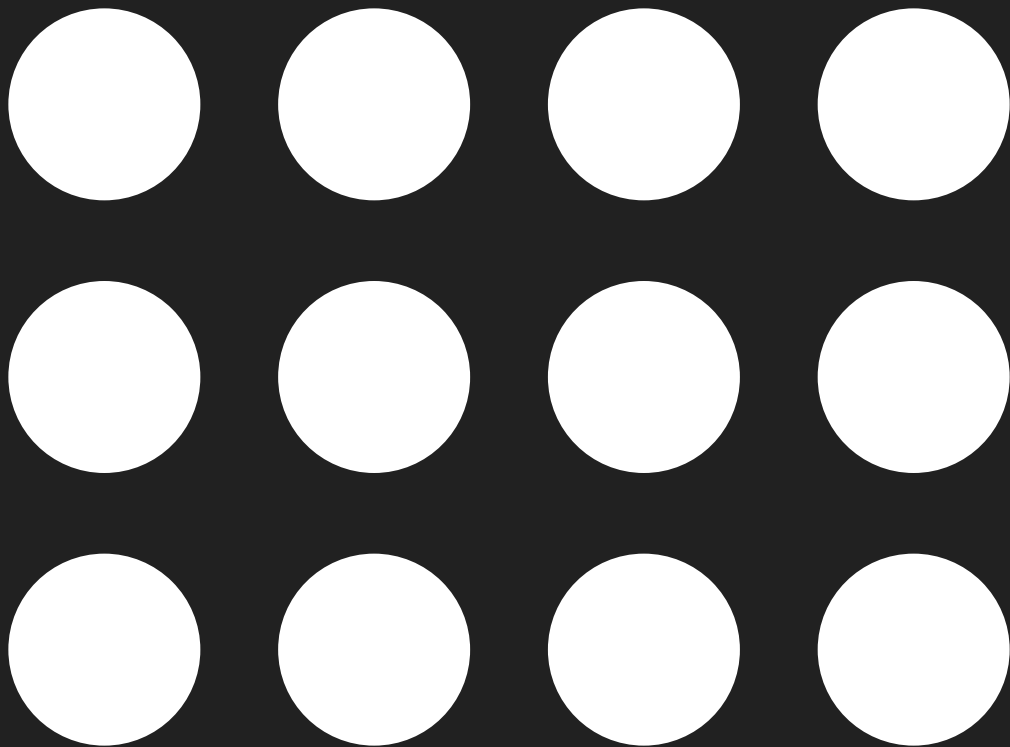
Year 0



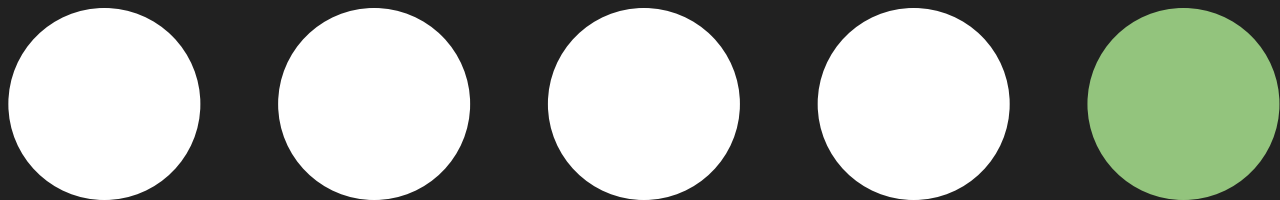
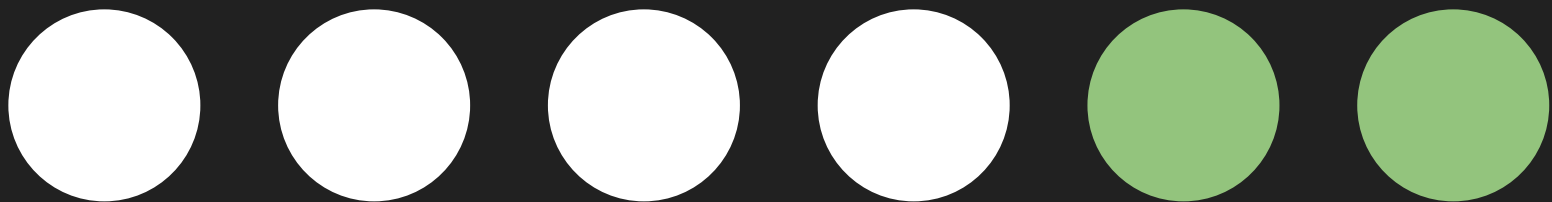
Year 0



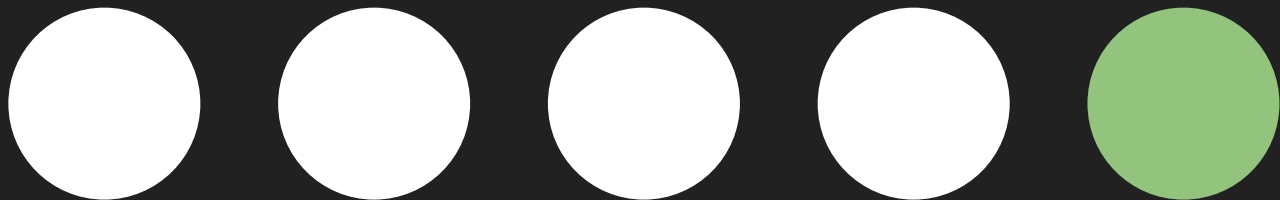
Year 0

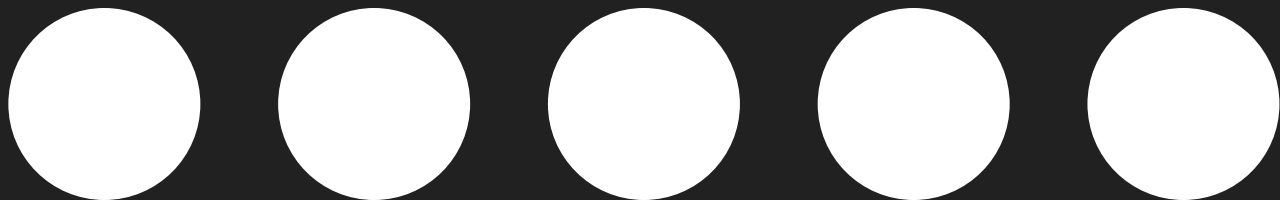


Year 0

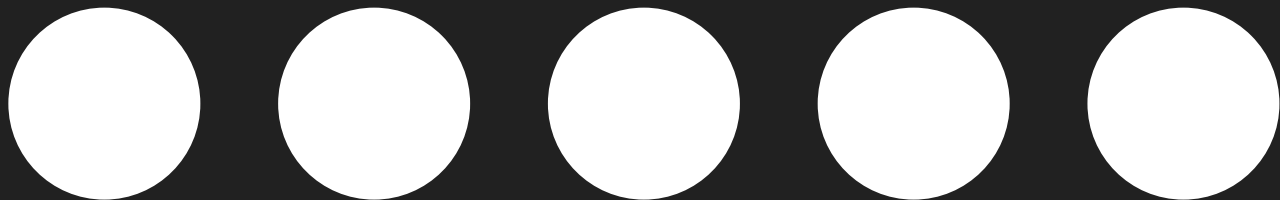


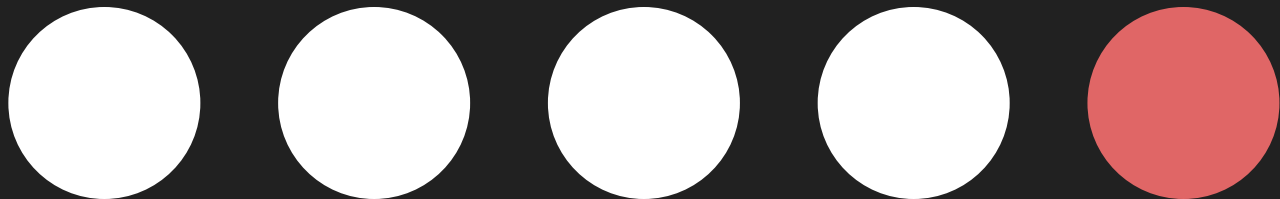
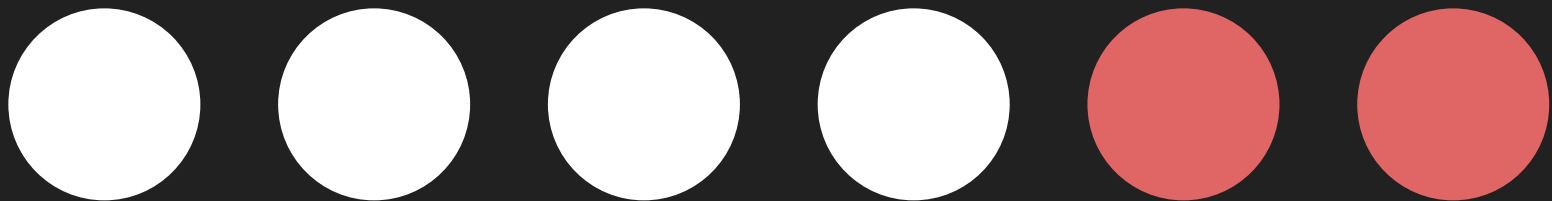
Year 1



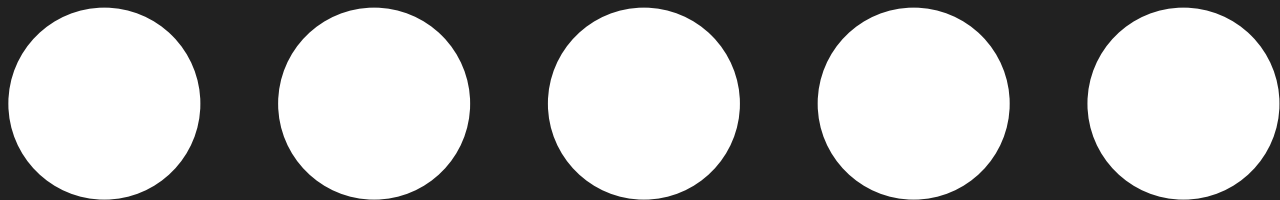


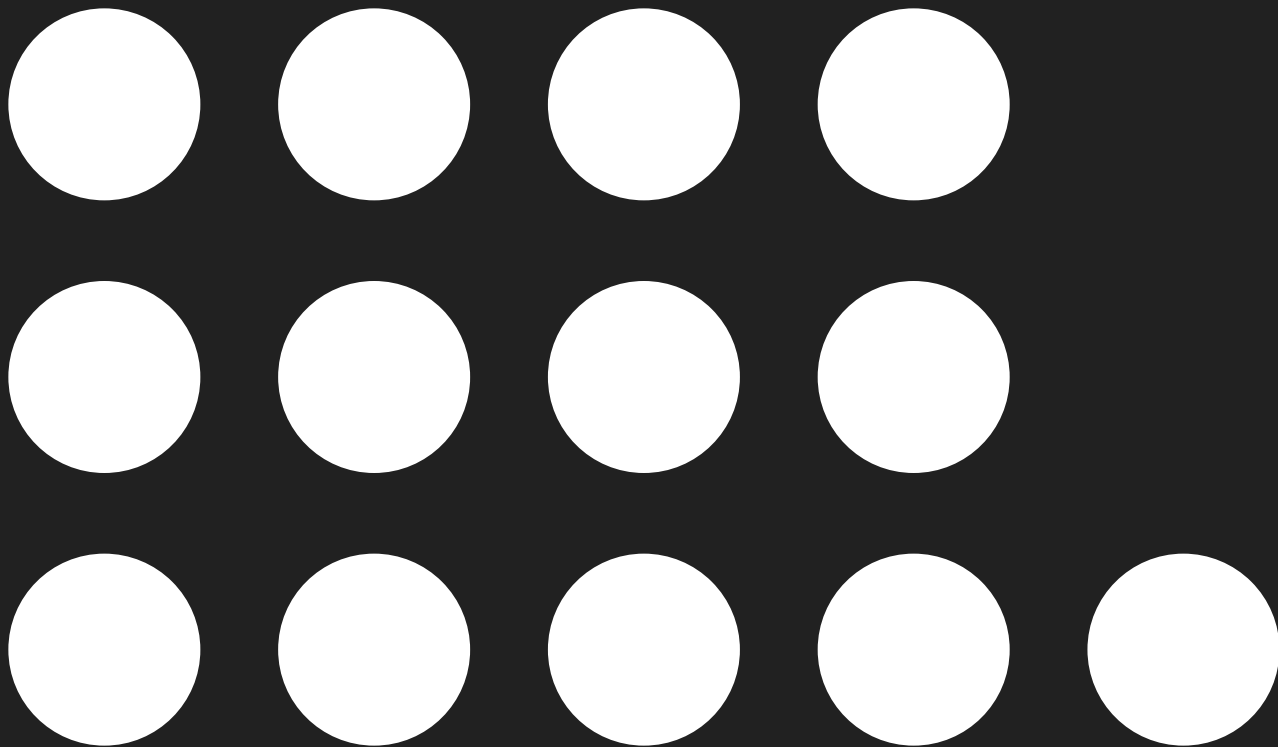
Year 1





Year 1





Year 1

We have a population of **12** llamas.
Each year, **$12/3$** new llamas are born,
and **$12/4$** llamas pass away.
How many years will it take to have a
population of **24** llamas?

- Work an example yourself
- Write down exactly what you did
- Create a generalization (algorithm) after working multiple examples
- Test your algorithm by hand
- Translate your algorithm to code
- Find bugs in your code by running test cases
- Debug (and critique) your code

Tutorials

Office Hours

cs50.ly/studybuddy

cs50.ly/attend

The background of the image is a light-colored surface covered with numerous colorful sticky notes. The notes are in various colors including pink, yellow, teal, and red. Each note features a hand-drawn sketch or doodle. Some sketches include faces, stars, geometric shapes like triangles and pentagrams, and abstract lines. One note has the word 'awesome' written on it. The central text 'This was CS50.' is overlaid on the sticky notes.

This was CS50.