Algorithms in C++: Assignment 3

1. Objective

Your goal is to write a program that will give you practice with recursion and vectors in C++.

2. Problem

You have been asked to find the number of different ways to climb an *n*-stair staircase. You are allowed to move up 1, 2, or 3 stairs at a time. You must also output how to climb all *n* stairs. For instance, there are 4 ways to climb 3 stairs, as seen below:

```
4 ways to climb 3 stairs.
1. [1, 1, 1]
2. [1, 2]
3. [2, 1]
4. [3]
```

Here are some examples of how the program should run:

Case 1: No input arguments
Print usage message.

```
$ ./stairclimber
Usage: ./stairclimber <number of stairs>
```

Case 2: Too many input arguments
Print usage message.

```
$ ./stairclimber 3 4 5
Usage: ./stairclimber <number of stairs>
```

Case 3: Bad input

```
$ ./stairclimber Howdy
Error: Number of stairs must be a positive integer.
```

Case 4: Bad input

```
$ ./stairclimber 0
Error: Number of stairs must be a positive integer.
```

Case 5: Valid

```
$ ./stairclimber 1
1 way to climb 1 stair.
1. [1]
```

Case 6: Valid ***(Notice the labels are right-aligned to the width of the highest label. You should do the same for labels that go up to any size.)***

```
$ ./stairclimber 5
13 ways to climb 5 stairs.
 1. [1, 1, 1, 1, 1]
 2. [1, 1, 1, 2]
 3. [1, 1, 2, 1]
 4. [1, 1, 3]
```

```
 5. [1, 2, 1, 1]
 6. [1, 2, 2]
 7. [1, 3, 1]
 8. [2, 1, 1, 1]
 9. [2, 1, 2]
10. [2, 2, 1]
11. [2, 3]
12. [3, 1, 1]
13. [3, 2]
```

3. Advice

Start this project early. Even though there aren't that many lines of code in the solution, working with recursion and nested vectors can get tricky.