# General Statement

This assignment is the first of a series. At this first step, it is an exercise in **straight-line code**. *It is intended to be tedious and repetitive*; it is part of a methodology I call *Cause Pain, Relieve Pain*. With each new programming feature that we learn, what is painful to do in the previous assignment is made easier by the new language feature that we are learning.

Therefore, if you happen to know a way to “do this better” using decisions, repetitions, lists, or other features from later chapters, **resist the urge**. You will have plenty of time to show off what you know in other situations.

## The Task

Simulate the operation of an elevator moving from the first to the 15th floor.

## Design

This is the Pencil and Paper step; it is the Top Level Design

1. Set up the simulation
2. At each floor, some passengers get off, some passengers get on, and then the elevator rises to the next floor.

## Refinement:

This is the detailed Pencil and Paper step; the Top Level has been both elaborated and “Divide-And-Conquered”

1. Set up the simulation
   1. Elevator is at the first floor
   2. Elevator has no passengers
2. At the first floor
   1. The elevator bell dings
   2. The floor is announced
   3. Some passengers get off
   4. Some passengers get on
   5. The elevator goes up to the next floor
3. At the second floor
   1. The elevator bell dings
   2. The floor is announced
   3. Some passengers get off
   4. Some passengers get on
   5. The elevator goes up to the next floor

… and so on, through 15 floors.

These statements translate almost directly into pseudocode commands. The numbered lines (1, 2, 3) will be comments, the lettered lines (a, b, c) will be statements/commands.

## Algorithm

1. Set up the simulation
   1. Set the elevator capacity to 20
   2. Set the floor to 1
   3. Set the number of passengers to 0
2. At the first floor
   1. Display “ding”
   2. Display “First floor”
   3. Set passengers = passengers – gettingOff1
   4. Set passengers = passengers + gettingOn1
   5. Set floor = floor + 1
3. At the second floor
   1. Display “ding”
   2. Display “Second floor”
   3. Set passengers = passengers – gettingOff2
   4. Set passengers = passengers + gettingOn2
   5. Set floor = floor + 1

… and so on, through 15 floors.Use the values in this table in your calculations

|  |  |  |
| --- | --- | --- |
| **Floor** | **Passengers Getting Off** | **Passengers Getting On** |
| 1 | 0 | 15 |
| 2 | 7 | 2 |
| 3 | 2 | 3 |
| 4 | 5 | 5 |
| 5 | 3 | 10 |
| 6 | 0 | 2 |
| 7 | 0 | 0 |
| 8 | 3 | 0 |
| 9 | 0 | 2 |
| 10 | 4 | 5 |
| 11 | 5 | 0 |
| 12 | 3 | 6 |
| 14 | 17 | 1 |
| 15 | 2 | 0 |

## Python Pseudocode Program Template

1. Start with the *Bare Minimum with Statement Block comments* template (the second one in the document).
2. Replace the < statement block n > placeholders with the comments (numbered lines above)
3. Insert the pseudocode commands between the comments.
4. Observe indentation and other style conventions
5. Use ***Turning Algorithm Design into Pseudocode*** as your guide