

3. What sequence of numbers would be printed by the following recursive procedure if we started it with N assigned the value 1?

```
procedure Exercise (N)
  print the value of N;
  if (N < 3) then
    (apply the procedure Exercise to the value N + 1);
  print the value of N.
```

4. What is the termination condition in the recursive procedure of Question/Exercise 3?

A short explanation on recursion.

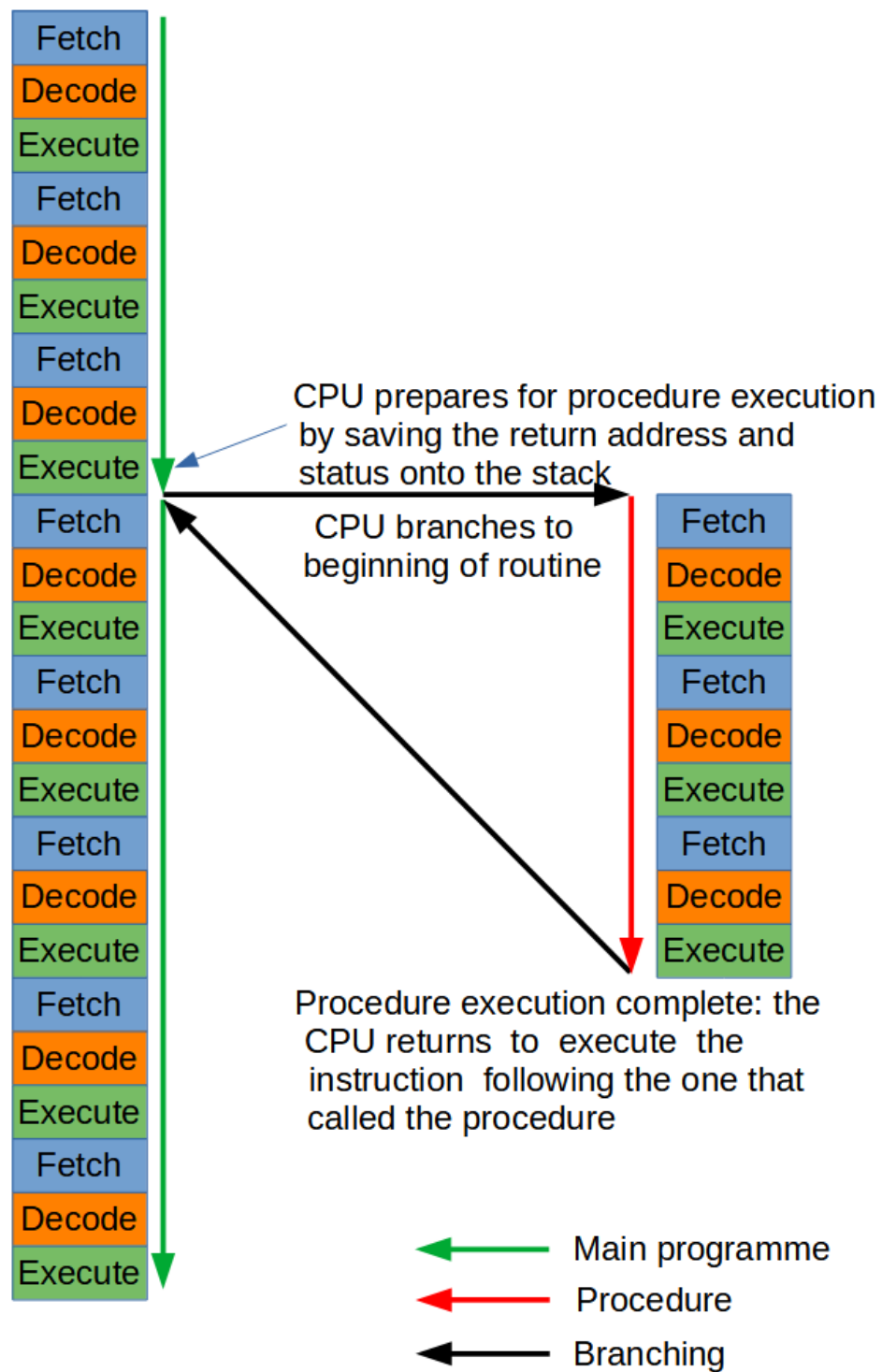
We attempt to solve problems by breaking them down into smaller ones, which we then attempt to solve one by one, by repeatedly calling a function. **Recursion** is the iterative process within which the function that solves the smaller problems, **calls itself**. This function is called a **recursive** function and it calls itself repeatedly until the **base case** is reached, when the **termination condition** evaluates to true. The solution of the base case is provided by the author of the programme.

In this exercise, the **recursive function** is the **procedure Exercise** and the **base case** is encountered when **N stops being less than 3**.

What happens within a computer when calling a procedure.

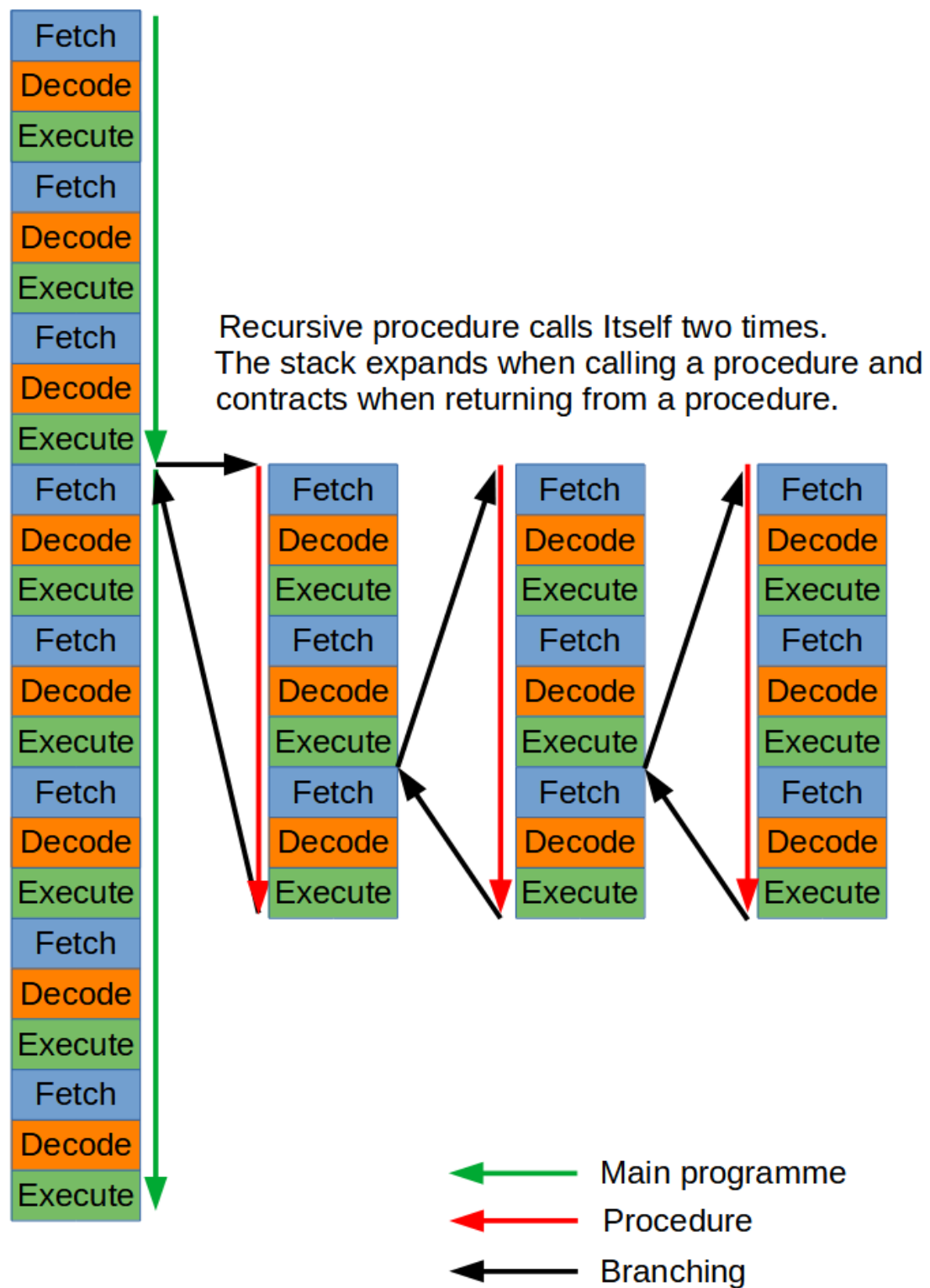
Main programme

Procedure routine



What happens within a computer when calling a procedure recursively.

Main programme



If the base case is not well defined, the procedure will continue to call itself indefinitely, gradually pushing more and more return and status information onto the stack. There will be a point that the available stack memory will be exhausted and the computer will crash.

The following table summarises what takes place as procedure Exercise is called recursively in the given exercise.

Step	Action	Stack content
1	Main programme assigns 1 to N	Stack is empty
2	Procedure Exercise (N) is called with N = 1	1. Return address to main programme
3	The number 1 is printed on the screen	
4	N < 3 => Procedure Exercise (N) is called with N = 2	2. Return address to Exercise(N) ₁ 1. Return address to main programme
5	The number 2 is printed on the screen	
6	N < 3 => Procedure Exercise (N) is called with N = 3	3. Return address to Exercise(N) ₂ 2. Return address to Exercise(N) ₁ 1. Return address to main programme
7	The number 3 is printed on the screen	
8	N = 3 => Procedure Exercise (N) is not called again	
9	The number 3 is printed on the screen	
10	Procedure Exercise (N) returns to its caller	2. Return address to Exercise(N) ₁ 1. Return address to main programme
11	The number 2 is printed on the screen	
12	Procedure Exercise (N) returns to its caller	1. Return address to main programme
13	The number 1 is printed on the screen	
14	Procedure Exercise (N) returns to its caller	Stack is empty
15	Main programme continues its execution	

The following numbers are printed during this process:

1, 2, 3, 3, 2, 1