**Algorithms**

1. a) Run this [flow chart example](https://dmaccarthy.github.io/sci/cs_new/ct1/img/flow1.svg) with yourself as the “computer.” Use the tables provided below to record the algorithm’s output and to keep track of the working memory. Run the algorithm twice with different input: 75 the first time and 360 the second time.

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| 75 | |  |
|
| **Working Memory** | |
| num |  |
| factor |  |

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| 360 | |  |
|
| **Working Memory** | |
| num |  |
| factor |  |

b) What does this algorithm do? How might this algorithm be useful?

Answer

2. a) Here is an algorithm described using *pseudocode*.

**INPUT a positive integer as num1**

**INPUT a positive integer as num2**

**SET small = minimum of num1 and num2**

**SET big = maximum of num1 and num2**

**SET gcf = small**

**WHILE big ÷ gcf has a remainder OR small ÷ gcf has a remainder...**

**SET gcf = gcf - 1**

**END-WHILE**

**PRINT gcf**

Run the algorithm with yourself as the “computer.” Use the tables provided on the next page to record the algorithm’s output and to keep track of the working memory. Run the algorithm twice with different input: 13 and 5 the first time and 10 and 15 the second time.

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| 13  5 | |  |
|
| **Working Memory** | |
| num1 |  |
| num2 |  |
| small |  |
| big |  |
| gcf |  |

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| 10  15 | |  |
|
| **Working Memory** | |
| num1 |  |
| num2 |  |
| small |  |
| big |  |
| gcf |  |

b) What does this algorithm do? How might this algorithm be useful?

Answer