*Simpson College*

*Computer Science Club*

Programming Challenges

Fall 2016

**All problems will use the standard input (keyboard) and output (console).**

**Each program must complete within 60 seconds.**

Fibonacci Variations

Fibonacci numbers are integers which fall in the Fibonacci sequence, a sequence characterized by the fact that each number in the sequence is the sum of the two preceding numbers:





By definition, the two starting numbers are 0, and 1 respectively.



However, if we alter these starting numbers we can attain variations on the Fibonacci sequence. For example, if we set we get this sequence of numbers:



Your task is to find all of the variations of that would produce a sequence that includes the input number, This is the rendered form of the equation. You can not edit this directly. Right click will give you the option to save the image, and in most browsers you can drag the image onto your desktop or another program., excluding where .

|  |  |
| --- | --- |
| **Input** | **Output** |
| The input will consist of multiple lines with a single number on each (no more than 1000 lines) with the last line consisting of the number 0. Do not process this line.  All numbers will be greater than 1 (except the last line) and less than or equal to 10,000,000. | The variations of that create a Fibonacci sequence that contains the input number, This is the rendered form of the equation. You can not edit this directly. Right click will give you the option to save the image, and in most browsers you can drag the image onto your desktop or another program., for that line, or if there are no sequences that produce This is the rendered form of the equation. You can not edit this directly. Right click will give you the option to save the image, and in most browsers you can drag the image onto your desktop or another program.then express this.  **If there are sequences that match:**  *N* can be found where F1 = *#1*, *#2*, *#3*, ...  With *N* being replaced with the input number, and the numbers (*#1, #2, etc)* replaced with the starting that would contain This is the rendered form of the equation. You can not edit this directly. Right click will give you the option to save the image, and in most browsers you can drag the image onto your desktop or another program.  **If there are NO sequences that match:**  *N* cannot be found in any variation of F1 |
| **Example**   |  |  | | --- | --- | | 8  98  2437  1230  0 | 8 can be found where F1 = 1, 4  98 can be found where F1 = 49  2437 cannot be found in any variation of F1  1230 can be found where F1 = 246, 410, 615 | | |

Hardy-Ramanujan Numbers

On a visit to the Indian mathematician S Ramanujan, British mathematician G H Hardy remarked that he had ridden in taxicab number 1729 and said that the number seemed to him rather a dull one. "No," Ramanujan replied, "it is a very interesting number; it is the smallest number expressible as the sum of two cubes in two different ways." This is the Hardy-Ramanujan number.

Your task is to find other numbers that fit this pattern; write a program to find the first This is the rendered form of the equation. You can not edit this directly. Right click will give you the option to save the image, and in most browsers you can drag the image onto your desktop or another program.smallest positive integers , including 1729, such that:



where  are positive integers, with .

The program must display  for the first This is the rendered form of the equation. You can not edit this directly. Right click will give you the option to save the image, and in most browsers you can drag the image onto your desktop or another program.smallest positive integers 

|  |  |
| --- | --- |
| **Input** | **Output** |
| The input will consist of a single integer, This is the rendered form of the equation. You can not edit this directly. Right click will give you the option to save the image, and in most browsers you can drag the image onto your desktop or another program.representing how many elements inside of the Hardy-Ramanujan sequence your program must find. | For each Hardy-Ramanujan number output  on its own line. Then:  If  print:  [*a*, *b*]  [*x*, *y*]  Else print:  [*x*, *y*]  [*a*, *b*]  When printing the pairs ( or ), always print the smaller of the two numbers first.  Eg, if  you should print:  [*y, x*] Instead of [*x*, *y*] |
| **Example**   |  |  | | --- | --- | | 2 | 1729  [9, 10]  [1, 12]  4104  [10, 15]  [2, 16] | | |

Simpson College Logo

Simpson College would like a piece of software that would display their logo in ASCII text at varying sizes, using asterisks. Your task is to write a program that will take the size of the logo as input and display ‘SC’ with the specified character dimensions.

|  |  |
| --- | --- |
| **Input** | **Output** |
| The input will consist of several odd (mathematically) integers, which will be greater than or equal to 5, and less than or equal to 55. The last line of the input will be ‘0’. Do not process this line. | The output should appear in the same order as the inputs. Each output should start with the line  “Case c: Dimension s”  (where c is the number of the input and s is the value of the input (the odd integer)), followed by  “SC”,  of height and width s, printed in stars. There should be one space between the “S” and the “C”. There should be a blank line between outputs. |
| **Example**   |  |  | | --- | --- | | 5  7  0 | Case 1: Dimension 5  \*\*\*\*\* \*\*\*\*\*  \* \*  \*\*\*\*\* \*  \* \*  \*\*\*\*\* \*\*\*\*\*  Case 2: Dimension 7  \*\*\*\*\*\*\* \*\*\*\*\*\*\*  \* \*  \* \*  \*\*\*\*\*\*\* \*  \* \*  \* \*  \*\*\*\*\*\*\* \*\*\*\*\*\*\* | | |

String Arithmetic

Your task is to build a simple mathematical operations calculator. Your program will receive an input with a mathematical expression using only the four basic operators , with no parentheses. Your program should determine the result of the expression.

All operations shall remain as integers (eg. ), and the operations will be calculated from left to right, but giving precedence to multiplication/division above addition/subtraction. Therefore:









|  |  |
| --- | --- |
| **Input** | **Output** |
| The input will consist of multiple lines with a mathematical expression on each (no more than 1000 lines) with the last line consisting of the number 0. Do not process this line.  Only positive numbers will be given, with no white space, but remember that subtraction can still cause numbers to become negative. | Your program should output the expression received, followed by an ‘ = ’ then the calculated value of the expression |
| **Example**   |  |  | | --- | --- | | 1/2  1+2-3/4\*5  0-10+1/2  0 | 1/2 = 0  1+2-3/4\*5 = 3  0-10+1/2 = -10 | | |

Sum Of Digits

Professor Brovenapova loves number games. One game the Professor likes is finding the sum of the squares of all the digits of a number. If you do this repeatedly, you will ultimately get to either one or four. For example the number 16:

16 becomes

37 (12 + 62) which becomes

58 (32+72) which becomes

89 (52+82), becoming

145 (82+92), becoming

42 (12+42+52), becoming

20 (42+22), and

4 (22+02), in succession.

|  |  |
| --- | --- |
| **Input** | **Output** |
| The input will consist of multiple lines with a integer on each line (less than 10,000,00), with the last line containing a 0. Do not process this last line. There will not be any more than 1,000 lines. | The output should appear in the same order as the inputs. Each output should be of the form “Case c” (where c is the number of the input) followed on successive lines by the sequence of digit-square sums, beginning with the input value and ending with 1 or 4. Output cases should be separated by blank lines. |
| **Example**   |  |  | | --- | --- | | 1  4194304  0 | Case 1  1  1  Case 2  4194304  139  91  82  68  100  1 | | |