Computer Programming I — Homework Assignment #3

**1.** Write a program that finds the two largest values among several integers. Assume that the first integer read specifies the number of values remaining to be entered. The screen dialog should appear as follows:

|  |
| --- |
| **Enter the number of integers to be processed followed by the integers:**  **6 10 3 15 21 26 14**  **Largest is 26**  **Second largest is 21**  **請按任意鍵繼續 . . .** |

**2.** (*UVa Online Judge: 11332*) For a positive integer *n*, let *f*(*n*) denote the sum of the digits of *n* when represented in base 10. It is easy to see that the sequence of numbers *n*, *f*(*n*), *f*(*f*(*n*)), *f*(*f*(*f*(*n*))), . . . eventually becomes a single digit number that repeats forever. Let this single digit be denoted *g*(*n*).

For example, consider *n*  1234567892.

Then:

*f*(*n*)  1234567892  47

*f*(*f*(*n*))  4  7  11

*f*(*f*(*f*(*n*)))  1  1  2

Therefore, *g*(1234567892)  2. The screen dialog should appear as follows:

|  |
| --- |
| **Enter a positive integer: 11**  **2**  **請按任意鍵繼續 . . .** |

|  |
| --- |
| **Enter a positive integer: 47**  **2**  **請按任意鍵繼續 . . .** |

|  |
| --- |
| **Enter a positive integer: 1234567892**  **2**  **請按任意鍵繼續 . . .** |

**3.** (*UVa Online Judge: 256*) The number 3025 has a remarkable quirk: if you split its decimal representation in two strings of equal length (30 and 25) and square the sum of the numbers so obtained, you obtain the original number:

(30  25)2  3025

The problem is to determine all numbers with this property having a given even number of digits.

For example, 4-digit numbers run from 0000 to 9999. Note that leading zeroes should be taken into account. This means that 0001 which is equal to (00  01)2 is a quirksome number of 4 digits. The number of digits may be 2, 4, 6 or 8. The screen dialog should appear as follows:

|  |
| --- |
| **Enter the number of digits (taken from 2, 4, 6, 8): 2**  **00**  **01**  **81**  **請按任意鍵繼續 . . .** |

|  |
| --- |
| **Enter the number of digits (taken from 2, 4, 6, 8): 6**  **000000**  **000001**  **088209**  **494209**  **998001**  **請按任意鍵繼續 . . .** |

**4.** An integer is said to be a *perfect number* if the sum of its proper factors, including 1 (but not the number itself), is equal to the number. For example, 6 is a perfect number, because 6 . Write a program that reads in a positive integer *n* and prints all the perfect numbers between 1 and *n*. Print the factors of each perfect number to confirm that the number is indeed perfect. The screen dialog should appear as follows:

|  |
| --- |
| **Enter a positive integer: 1000**  **Perfect numbers between 1 and 1000:**  **6 = 1 + 2 + 3**  **28 = 1 + 2 + 4 + 7 + 14**  **496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248**  **請按任意鍵繼續 . . .** |

**5**. Write a program that prints the multiplication table as follows:

|  |
| --- |
| **2 \* 1 = 2 3 \* 1 = 3 4 \* 1 = 4 5 \* 1 = 5**  **2 \* 2 = 4 3 \* 2 = 6 4 \* 2 = 8 5 \* 2 = 10**  **2 \* 3 = 6 3 \* 3 = 9 4 \* 3 = 12 5 \* 3 = 15**  **2 \* 4 = 8 3 \* 4 = 12 4 \* 4 = 16 5 \* 4 = 20**  **2 \* 5 = 10 3 \* 5 = 15 4 \* 5 = 20 5 \* 5 = 25**  **2 \* 6 = 12 3 \* 6 = 18 4 \* 6 = 24 5 \* 6 = 30**  **2 \* 7 = 14 3 \* 7 = 21 4 \* 7 = 28 5 \* 7 = 35**  **2 \* 8 = 16 3 \* 8 = 24 4 \* 8 = 32 5 \* 8 = 40**  **2 \* 9 = 18 3 \* 9 = 27 4 \* 9 = 36 5 \* 9 = 45**  **6 \* 1 = 6 7 \* 1 = 7 8 \* 1 = 8 9 \* 1 = 9**  **6 \* 2 = 12 7 \* 2 = 14 8 \* 2 = 16 9 \* 2 = 18**  **6 \* 3 = 18 7 \* 3 = 21 8 \* 3 = 24 9 \* 3 = 27**  **6 \* 4 = 24 7 \* 4 = 28 8 \* 4 = 32 9 \* 4 = 36**  **6 \* 5 = 30 7 \* 5 = 35 8 \* 5 = 40 9 \* 5 = 45**  **6 \* 6 = 36 7 \* 6 = 42 8 \* 6 = 48 9 \* 6 = 54**  **6 \* 7 = 42 7 \* 7 = 49 8 \* 7 = 56 9 \* 7 = 63**  **6 \* 8 = 48 7 \* 8 = 56 8 \* 8 = 64 9 \* 8 = 72**  **6 \* 9 = 54 7 \* 9 = 63 8 \* 9 = 72 9 \* 9 = 81** |