1. Print numbers from 10 to 1.
2. Given a number print its digits count.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 78756 | 5 |
| -122 | 3 |
| 8 | 1 |
| 0 | 1 |

1. Count numbers of digit *9* in a number.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 129 | 1 |
| 929 | 2 |
| 192999 | 4 |
| 887 | 0 |

1. Given two numbers. Print powers of *2* between that numbers. (without using *Math.pow*)

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7, 45 | 8, 16, 32 |
| 0, 150 | 1, 2, 4, 8, 16, 32, 64, 128 |

1. Given a number as input, insert dashes (-) between each two even numbers.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 25468 | “254-6-8” |
| 7180322 | “718-032-2” |
| 42 | “4-2” |
| 222 | “2-2-2” |

1. An Armstrong number is a *3 digit* number for which sum of cube of its digits is equal to the number itself. Write a program to check whether the given number is an Armstrong number.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 124 | “No” |
| 153 | “Yes” |
| 371 | “Yes” |

1. Given a positive number. Print it in reverse order.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1253 | 3521 |
| 122 | 221 |

1. Insert a number. Print “yes” if it contains *3* successive zeros, otherwise print “no”.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10154 | “no” |
| 350120003 | “yes” |
| 120900 | “no” |

1. A perfect number is a positive integer that is equal to the sum of its proper positive divisors. Print all perfect numbers between *1* and *500*.
2. Insert a number. Remove all zeros from the number, except the last one and print the number. If there are at most one zero, print “Nothing to remove”.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4050120 | 45120 |
| 7845012 | “Nothing to remove.” |
| 0 | “Nothing to remove.” |

1. Given a number *n (0 ≤ n ≤ 100)* you should print the smallest positive number such that its digits product is equal to*n***.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 | 25 |
| 18 | 29 |
| 81 | 99 |

1. Insert a *n* positive number. Print the *n*-st prime number.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 11 |
| 14 | 43 |

1. Insert two positive integers *n* and *m* between *1* and *10*. Compute and print *nm*.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4, 5 | 1024 |
| 6, 10 | 60466176 |

1. Find the biggest common divisor of two inserted numbers.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6, 10 | 2 |
| 12, 24 | 12 |
| 78, 7 | 1 |

1. Given a number. Print all prime factors of the number.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 420 | “2, 3, 5, 7” |
| 99 | “3, 11” |

1. Enter a positive number. Calculate and print its factorial. ( n! = n \* (n-1) \* (n-2) \* … \* 3 \* 2 \* 1 , 0! = 1 )

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | “5! = 120” |
| 1 | “1! = 1” |
| 7 | “7! = 5040” |

1. Enter a number. Reverse its first and last digits. Print the new number.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | 2 |
| 13 | 31 |
| 895796 | 695798 |

1. Given a number ***n ( n >= 0 )***. Print ***n*** Fibonacci number. *(Fibonacci series: 0, 1, 1, 2, 3, 5, 8 …,* ak = ak-1 + ak-2)

|  |  |
| --- | --- |
| **Input** | **Output** |
| 0 | 0 |
| 2 | 1 |
| 10 | 55 |
| 20 | 6765 |

1. Given a number ***n* *( n > 0 )***. Print Fibonacci series up to ***n***.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7 | “0, 1, 1, 2, 3, 5 “ |
| 45 | “0, 1, 1, 2, 3, 5, 8, 13, 21, 34” |

1. Print the following number pattern:

1  
 12  
 123  
 1234  
 12345  
 1234  
 123  
 12  
 1

1. Write a program, to calculate the value of the following sequence:  
   *1 - 1/3 + 1/5 - 1/7 + 1/9 + ….. + \* 1/n .*
2. Write a program which will give you all of the potential combinations of a two-digit decimal combination, printed in a comma delimited format

‘00’, ‘01’, ‘02’, …, ‘98’, ‘99’

1. Write a program to check the validity of password input by users.

Validation:

* At least 1 letter between [a-z] and 1 letter between [A-Z].
* At least 1 number between [0-9].
* At least 1 character from [$#@].
* Minimum length 6 characters.
* Maximum length 16 characters.

|  |  |
| --- | --- |
| **Input** | **Output** |
| “12asdf” | “Invalid” |
| “Aaza1234566#” | “Valid” |

1. Write a program to print X star pattern series using loop.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | \* \*  \* \*  \* \*  \* \*  \*  \* \*  \* \*  \* \*  \* \* |