ASSUMPTION UNIVERSITY

Vincent Mary School of Science and Technology

Department of Computer Science

CS3201

Algorithm Design

Term Project Report

1014. Product of Digits

Timus Online Judge

Submit to

Asst. Prof. Dr. Thitipong Tanprasert

by

5810127 Thanawadee Khemasrisuwan

5811095 Varissara Tangsajjanuraksa

Semester 1/2017

INTRODUCTION

Problem: 1014. Product of Digits

Time Limit: 1.0 second

Memory Limit: 64 MB

Difficulty: 104

Description:

Your task is to find the minimal positive integer number Q so that the product of digits of Q is exactly equal to N.

Input:

The input contains the single integer number $N(0 \leq N \leq 10^9)$.

Output:

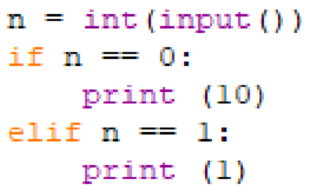
Your program should print to the output the only number Q . If such a number does not exist print -1.

Sample input and output

Input	Output
10	25

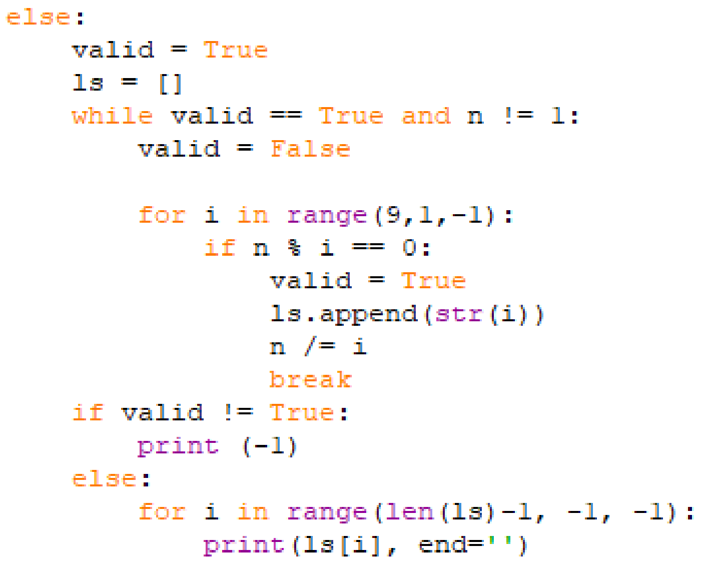
Problem Source: Ural State University Internal Contest ،99 #2

Problem Solution



```
n = int(input())
if n == 0:
    print (10)
elif n == 1:
    print (1)
```

First, we received the integer input and assign it to variable. If the input equals to zero, the result will be 10 as the product of digits (1 and 0) equals to 0 and it is the smallest value possible. In case where input equals to 1, 1 would be the smallest value itself.



```
else:
    valid = True
    ls = []
    while valid == True and n != 1:
        valid = False

        for i in range(9,1,-1):
            if n % i == 0:
                valid = True
                ls.append(str(i))
                n /= i
                break
    if valid != True:
        print (-1)
    else:
        for i in range(len(ls)-1, -1, -1):
            print(ls[i], end='')
```

In the else part, we use valid variable to indicate whether there is at least one possible value that has product of digits equals to input or not. For example, if the input is prime number like 13, as 13 cannot be factorized any further. There is no possible output we can have. In this case, we printed -1.

Besides, we created a list to keep the number that we factorize it from the input. We would continue the loop while valid is true and n does not equal to 1 (number can be factorized further). We would begin the loop at i equals to 9 and end at i equals to 2 (loop in decreasing order). We would factorize input value by trying to find if it can be divided by i or not. If it is divisible by i, we appended the number i into ls list. As we divided from bigger number like 9 to another 1-digit smaller value. We got numbers in descending order such as 52 from input value 10.

After that we printed out the value from last element in ls to first element in ls. From the same example input 10, we printed out 2 and then 5. Then, we can get the smallest possible product of digits.

Test Case

#1

Input	Output
0	10

#2

Input	Output
1	1

#3

Input	Output
31	-1

#4

Input	Output
7	7

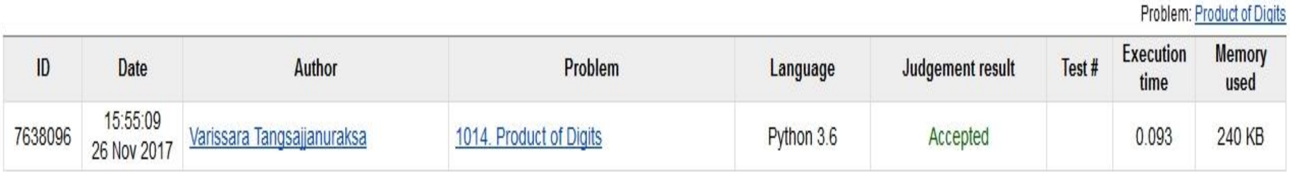
#5

Input	Output
1000000	55555588

#6

Input	Output
10000000000000000	2555555555555555588888

Submission Result



Problem: Product of Digits

ID	Date	Author	Problem	Language	Judgement result	Test #	Execution time	Memory used
7638096	15:55:09 26 Nov 2017	Varissara Tangsajjanuraksa	1014. Product of Digits	Python 3.6	Accepted		0.093	240 KB