

# Maximum Product

DiPS CodeJam 22

---

## Prompt

---

You are given a list of integers  $n_1 \dots n_k$  (not necessarily distinct) where each has a value between 1 and 9. Using each of  $n_1 \dots n_k$  exactly once, you can form concatenations of digits, to achieve a new list of numbers. You then have to output the product of this new list. The goal is to maximize this product by choosing the best concatenations of digits.

## Input Format

The first and only line of the input contains  $n_1 \dots n_k$ , space-separated.

## Output Format

The first and only line of your output must contain the maximum product.

## Constraints

- $1 \leq n \leq 9$
- $3 \leq k \leq 100$

## Sample Input/Output

Input	Output
3 5 5 2	2756

## Solution

---

The optimal solution can be found applying a single multiplication. This gives:

$$(10x + d)y > (10y + d)x$$

where  $x$  and  $y$  are the two numbers and  $d$  is the next digit to insert. It follows that  $y > x$  and thus each new digit must be concatenated to the smallest number. Digits should be added in descending order. Applying this to the sample input **3 5 5 2**, we get the following steps:

5, 0  
5, 5  
53, 5  
53, 52

The answer is  $53 \cdot 52$ , or 2756.

## Sample Program

---

```
l = sorted(map(int, input().split()), reverse=1)
m = [0,0]
for x in l:
    i = m[0] > m[1]
    m[i] = m[i]*10 + x
    # Uncomment the following line to see how it's concatenating the digits.
    # print(m)
print(m[0] * m[1])
```