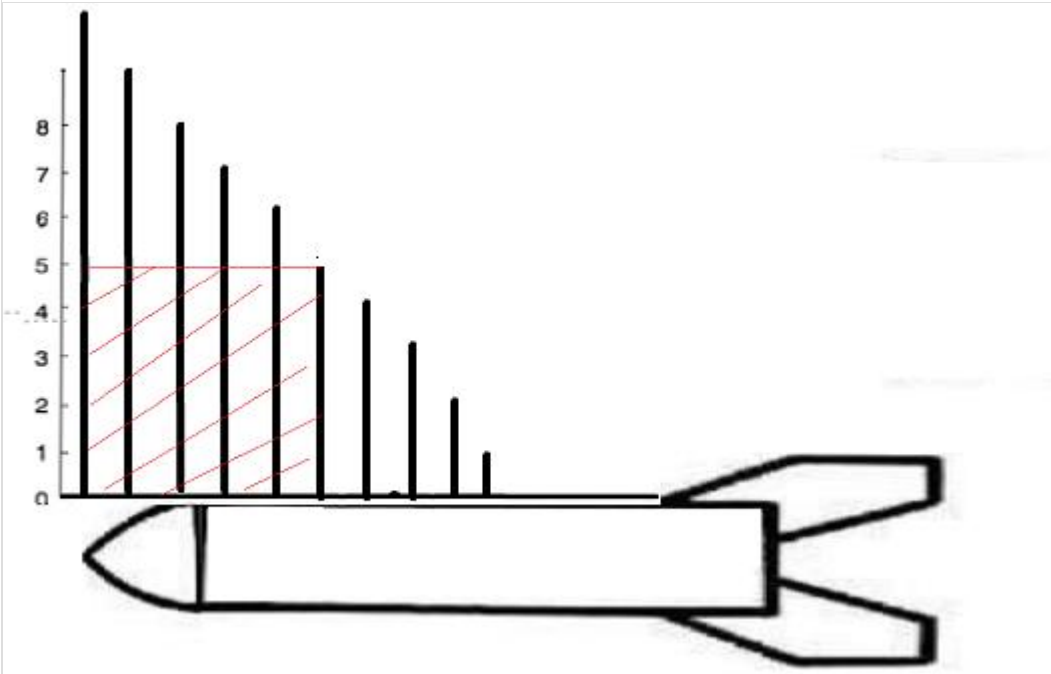


题目描述：

给航天器一侧加装长方形或正方形的太阳能板（图中的红色斜线区域），需要先安装两个支柱（图中的黑色竖条），再在支柱的中间部分固定太阳能板。但航天器不同位置的支柱长度不同，太阳能板的安装面积受限于最短一侧的那根支柱长度。如图：



现提供一组整形数组的支柱高度数据，假设每根支柱间距离相等为 1 个单位长度，计算如何选择两根支柱可以使太阳能板的面积最大。

输入描述：

10,9,8,7,6,5,4,3,2,1

注：支柱至少有 2 根，最多 10000 根，能支持的高度范围 $1\sim 10^9$ 的整数。柱子的高度是无序的，例子中递减只是巧合。

输出描述：

可以支持的最大太阳能板面积：（ 10 米高支柱和 5 米高支柱之间）

25

补充说明：

10 米高支柱和 5 米高支柱之间宽度为 5，高度取小的支柱高也是 5，面积为 25。任取其他两根支柱所能获得的面积都小于 25。所以最大的太阳能板面积为 25。

示例 1

输入：

10, 9, 8, 7, 6, 5, 4, 3, 2, 1

输出：

25

```
# line = input().strip()
# a = [int(x) for x in line.split(',')]

# n = len(a)
# right = [n] * n
# stack = []
# for index in range(n):
#     while stack and a[stack[-1]] > a[index]:
#         right[stack[-1]] = index
#         stack.pop()

#     stack.append(index)

# left = [-1] * n
# stack = []
# for index in range(n - 1, -1, -1):
#     while stack and a[stack[-1]] > a[index]:
#         left[stack[-1]] = index
#         stack.pop()

#     stack.append(index)

# res = 0
# for index in range(n):
#     res = max(res, a[index] * (index - left[index] - 1), a[index] * (right[index] - index - 1))

# print(res)
s = input().split(',')
n = len(s)
arr = [int(x) for x in s]
l, r = 0, n - 1
ans = 0
while l < r:
    ans = max(ans, min(arr[l], arr[r]) * (r - l))
```

```
    if arr[l] < arr[r]:  
        l += 1  
    else:  
        r -= 1  
print(ans)
```