

# monisai

题目名称	棍子	数组	十一	棋盘
英文名称	sticks	count	eleven	chess
源程序文件名	sticks.cpp	count.cpp	eleven.cpp	chess.cpp
输入文件名	sticks.in	count.in	eleven.in	chess.in
输出文件名	sticks.out	count.out	eleven.out	chess.out
时间限制	1s	0.3s	1s	1s
空间限制	256MB	256MB	256MB	256MB
比较方式	实数比较	全文比较	全文比较	全文比较

编译选项（未确定）：-lm -O2 -std=c++14

评测机型号：

# A.棍子(sticks)

有  $n$  根棍子，长为  $a_0, a_1 \dots a_{n-1}$ 。你要做至多  $C$  次操作，每次选择一根棍子，把它劈成两半，每根长度可以是任意正数（不一定是整数）。要求最大化第 $k$ 长的棍子的长度。

## 输入格式

第一行三个正整数  $n, C$  和  $k$ 。

第二行 $n$ 个正整数  $a_0, a_1 \dots a_{n-1}$ 。

## 输出格式

输出一个实数，表示最大的第 $k$ 长的长度。要求绝对或相对误差不超过  $10^{-9}$ 。

## Sample1

Input

```
2 1 1
5 8
```

Output

```
8.000000000000000
```

## Sample2

Input

```
2 1 3
5 8
```

Output

```
4.000000000000000
```

## Sample3

Input

```
3 2 5
1000000000 1000000000 1
```

Output

```
1.000000000000000
```

Sample4

Input

```
13 789 456
76 594 17 6984 26 57 9 876 5816 73 969 527 49
```

Output

```
34.9200000000000
```

样例解释

在样例一中，你可以什么都不干。如果把  $k$  改成 2，也是一样的。

在样例二中，你可以把长为 8 的劈成两个 4 的。

在样例三中，你可以把长为  $10^9$  的劈成三根，只要三根长度都不小于 1 就行了

数据范围

$1 \leq n \leq 50; 1 \leq a_i, C, k \leq 10^9; k \leq C + n$

- Subtask 1(20 pts): $k = 2$ 。
- Subtask 2(20 pts): $C = 10^9$ 。
- Subtask 3(60 pts):无特殊限制。

## B.数组(count)

请计算满足以下条件的正整数数组  $(a, b, c)$  的个数：

$$\min A \leq a \leq \max A$$

$$\min B \leq b \leq \max B$$

$$\min C \leq c \leq \max C$$

$$c = ab$$

### 输入格式

第一行六个正整数  $\min A, \max A, \min B, \max B, \min C, \max C$  。

### 输出格式

输出满足条件的数组个数。

### Sample1

Input

6 8 4 5 27 35

Output

4

### Sample2

Input

8176 184561 1348 43168 45814517 957843164

Output

2365846085

## 样例解释

在样例一中，满足条件的数组有：  $(6, 5, 30), (7, 4, 28), (7, 5, 35), (8, 4, 32)$ 。

数据范围

$1 \leq \min A \leq \max A \leq 10^9$ ；对B和C也是类似的。

Subtask 1(20 pts): $\max A, \max B, \max C \leq 5000$

Subtask 2(30 pts): $\min B = 1, \max B = 10^9$

Subtask 3(50 pts):无特殊限制。

C.十一(eleven)

你有  $n$  个数，它们都不含前导0。你想把它们排列，然后顺次拼接起来。问有多少种方案使结果能被 11 整除，输出答案对  $10^9 + 7$  取模的结果。注意数字间都是可区分的，所以共有  $n!$  种方案。

输入格式

第一行一个正整数  $n$ ，接下来一行  $n$  个正整数  $a_0, a_1, \dots, a_{n-1}$ 。注意这些数字可能很大，超出long long的范围。

输出格式

输出满足条件的方案数对  $10^9 + 7$  取模的结果。

Sample1

Input

```
3
58 2012 123
```

Output

```
2
```

Sample2

Input

```
4
1 1111 1 11
```

Output

```
24
```

Sample3

Input

```
6
983 4654 98 3269 861 30981
```

Output

```
96
```

Sample4

Input

```
20
193 8819 40676 97625892 5719 45515667 32598836 70559374 38756 724
93391 942068 506 901150 874 895567 7560480 7427691 799450 85127
```

Output

```
537147821
```

样例解释

在样例一中，满足条件的方案有：58|2012|123，2012|58|123。  
在样例二中，虽然所有方案拼接之后都是"8个1"，但我们需要区分它们。

数据范围

- $n \leq 50$ ；每个数字至多有 50 位。
- Subtask 1(20 pts): $n \leq 8$
- Subtask 2(40 pts):每个数字位数相同。
- Subtask 3(40 pts):无特殊限制。

# D.棋盘(chess)

你有一个  $8 \times 8$  的棋盘，上面有一些黑点。

你可以进行以下操作：选一个黑点，计算它到其他未移除黑点的曼哈顿距离的最大值，将其作为本次操作的代价。然后移除这个黑点。

你要用最小的代价移除这些黑点。

曼哈顿距离定义：设  $(x, y)$  表示第  $x$  行第  $y$  列的格子。两个格子  $(x_1, y_1), (x_2, y_2)$  的曼哈顿距离为  $|x_1 - x_2| + |y_1 - y_2|$ 。

## 输入格式

输入一个  $8 \times 8$  的字符矩阵，每个字符为 '#' 或 '.'，第  $i$  行  $j$  列为 '#' 则此处有黑点，否则没有。

## 输出格式

输出最小代价。

## Sample1

Input

```
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....
```

Output

```
0
```

## Sample2

Input

```
.....  
.....  
...#...  
.#.....  
.....#  
.....  
.....  
.....
```

Output

10

Sample3

Input

```
##...####  
#####..#  
..#.#...  
#...##.##  
.#.###.#  
#####.###  
#.#...#.  
##....#.
```

Output

275

样例解释

在样例二中，你可以先移除(4, 7)。

数据范围

- subtask 1(20 pts):至多有 9 个黑点。
- subtask 2(20 pts):至多有 19 个黑点。
- subtask 3(20 pts):至多有 29 个黑点。
- subtask 4(40 pts):无特殊限制。