

[编程|20分] 柠檬的选择

时间限制：C/C++ 1秒，其他语言 2秒

空间限制：C/C++ 262144K，其他语言 524288K

64bit IO Format: %lld

本题可使用本地IDE编码，不做跳出限制，编码后请点击“保存并调试”按钮进行代码提交。

■ 题目描述

小Q的面前有两排柠檬。

我们称两排柠檬分别为 A, B。A 排有 n 个柠檬，B 排则有 m 个柠檬。每个柠檬上带有一个数字。大小均在 -10^9 到 10^9 之间。

柠檬突然来了，看到小Q有这么多柠檬，她非常生气，于是拿走了 A 排柠檬中的一个柠檬。

小Q不知道柠檬拿走的是哪一个柠檬，但是他想知道，在最坏情况下，A 中剩余的某个柠檬带的数字与 B 中某个柠檬带的数字之积最大能是多少。

输入描述:

第一行输入两个数 $n, m (2 \leq n, m \leq 3 \times 10^5)$ 。

第二行输入 n 个整数 A_1, A_2, \dots, A_n 。

第三行输入 m 个整数 B_1, B_2, \dots, B_m 。

输出描述:

输出仅一行一个整数，表示最坏情况下，最大值可能是多少。

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
2 2
20 18
2 14
```

输出

复制

```
252
```

说明

最坏情况是20被拿走了，那么最大为 $18 * 14 = 252$ 。

示例2 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
5 3
-1 0 1 2 3
-1 0 1
```

输出

复制

```
2
```

说明

最坏情况是3被拿走了，那么最大为 $2 * 1 = 2$ 。

```
def lemon(a, b):
    a.sort()
    b.sort()
    res = max(a[1] * b[0], a[-2] * b[-1])

    if res <= 0:
        tmp = 1e9
        idx = -1
        for i in range(len(a)):
            if abs(a[i]) < tmp:
                tmp = abs(a[i])
                idx = i
        tmp = 1e9
        idx2 = -1
        for i in range(len(a)):
            if i != idx and abs(a[i]) < tmp:
                tmp = abs(a[i])
                idx2 = i
        for i in range(m):
            res = max(res, a[idx2] * b[i])
```

```

        return res

s = raw_input()
[n, m] = [int(i) for i in s.split(' ')]
s1 = raw_input()
a = [int(i) for i in s1.split(' ')]
s2 = raw_input()
b = [int(i) for i in s2.split(' ')]
print lemon(a, b)

'''
2 2
20 18
2 14
'''

```

[编程|20分] 数轴游戏

时间限制: C/C++ 1秒, 其他语言 2秒

空间限制: C/C++ 262144K, 其他语言 524288K

64bit IO Format: %lld

本题可使用本地IDE编码, 不做跳出限制, 编码后请点击“保存并调试”按钮进行代码提交。

■ 题目描述

课堂上, 小Q老师讲了关于数轴的知识, 小明和小红在数轴上玩起了游戏。首先给出一个长度为 n 的无重复元素的数组 a 。一开始, 小明从这个数组中选择两个不同的数字 x 和 y , 然后小明会放2颗石子在数轴上的 $|x|$ 点与 $|y|$ 点, 代表小明占领了两个点之间 (包括端点) 的区间; 然后, 小红会放2颗石子在数轴上的 $|x+y|$ 点与 $|x-y|$ 点, 代表小红占领了两个点之间的区间 (包括端点)。小红的好胜心比较强, 所以她希望她所占领的区间完全包括了小明所占领的区间 (包括端点), 但是 x 和 y 是由小明选择的, 所以小红想知道有多少种选法可以满足她的愿望。

注意 (x,y) 和 (y,x) 是同一种选法, 只被计算一次。

输入描述:

第一行一个数字 n , 代表数组的大小。

第二行 n 个整数 $a_1, a_2, a_3, \dots, a_n$, 表示数组 a , 每两个数字之间用一个空格分隔。

数据保证数组中的数字两两不同, 且

$2 \leq n \leq 2 * 10^5, -10^9 \leq a_i \leq 10^9$ 。

输出描述:

输出描述:

一个整数，表示满足条件的选法的个数。

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

2
1 5

输出

复制

0

说明

小明只能选择 (1, 5)，故小明的区间为 [1, 5]，小红的区间为 [4, 6]，不满足条件。

示例2 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

3
1 -2 2

输出

复制

3

```
def main(nums, n):
    nums.sort()
    res = 0
    for i in range(n-1):
        if nums[i+1] > nums[i] * 2:
            continue
        left = i + 1
        right = n - 1
        while left < right:
            mid = (left + right + 1) / 2
            if nums[mid] <= nums[i] * 2:
                left = mid
            else:
                right = mid - 1
        res += (right - i)
    print res

n = int(raw_input())
s = raw_input()
nums = [abs(int(i)) for i in s.split(' ')]
```

```
main(nums, n)
```

```
'''
```

```
2
```

```
1 5
```

```
'''
```

```
'''
```

```
3
```

```
1 -2 2
```

```
'''
```

[编程|20分] 游戏

时间限制：C/C++ 2秒，其他语言 4秒

空间限制：C/C++ 524288K，其他语言 1048576K

64bit IO Format: %lld

本题可使用本地IDE编码，不做跳出限制，编码后请点击“保存并调试”按钮进行代码提交。

■ 题目描述

小Q非常喜欢玩游戏。在某一天他在WeGame上购买了一些他感兴趣的游

戏。
他一共买了 n 个游戏，每个游戏有两个属性 x_i 和 y_i ，分别表示游戏包含的关卡的数量和游戏的难度。小Q打游戏所获得的快乐被定义为他玩的所有游戏的总关卡数量与所玩游戏中的最小难度的乘积。假设他玩了2个游戏，关卡数量和难度分别为[1,3]和[2,4]，那么他所获得的快乐就等于 $(1 + 3) * 2 = 8$ 。由于时间有限，现在小Q最多只能选择玩 k 个游戏，请你帮助他计算他能获得的最多的快乐。

输入描述:

第一行两个整数 n 和 k ;

接下来 n 行，第 i 行包含两个整数 x_i 和 y_i ;

满足 $1 \leq k \leq n \leq 3 * 10^5; 1 \leq x_i, y_i \leq 10^6$ 。

输出描述:

一个整数，表示小Q能获得的最多的快乐。

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
3 2
1 1
2 2
3 3
```

输出

复制

```
10
```

示例2 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
1 1
2 3
```

输出

复制

```
6
```

```
import heapq
def main(games, n, k):
    games = sorted(games, key=(lambda x: x[1]), reverse=True)

    s = 0
    res = 0
    heap = []
    for i in range(k):
        heap.append(games[i][0])
        s += games[i][0]

        res = max(res, s * games[i][1])
    heapq.heapify(heap)
    for i in range(k, n):
        s += games[i][0]
        s -= heapq.heappushpop(heap, games[i][0])

        res = max(res, s * games[i][1])

    print res

s = raw_input()
[n, k] = [int(i) for i in s.split(' ')]

games = []
```

```

for i in range(n):
    s = raw_input()
    [a, b] = [int(i) for i in s.split(' ')]
    games.append([a, b])

main(games, n, k)

'''
3 2
1 1
2 2
3 3
'''

```



示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```

4
10
114514 1919810 +
2
100 11 *
36
TXDY DJQ -
36
3414043104 998244535 -

```

输出

复制

```

2034324
1100
TJU8
2URVXZYVWZ

```

```

def func(n, k):
    a = range(10) + list(map(chr, range(65, 100)))
    b = []
    res = ''
    while True:
        x = n // k
        y = n % k
        b += [y]
        if x == 0:
            break
        n = x
    b = b[::-1]

```

```

    for i in b:
        res += str(a[i])
    return res

T = int(raw_input())
total = []
for i in range(T):
    K = int(raw_input())
    s = raw_input()
    [s1, s2, op] = s.split(' ')
    n1 = int(s1, base=K)
    n2 = int(s2, base=K)
    if op == '+':
        res = func(n1 + n2, K)
    elif op == '-':
        res = func(n1 - n2, K)
    elif op == '*':
        res = func(n1 * n2, K)
    total.append(res)

for t in total:
    print t

'''
4
10
114514 1919810 +
2
100 11 *
36
TXDY DJQ -
36
3414043104 998244535 -
'''

```


[编程|20分] 小Q取名字

时间限制：C/C++ 2秒，其他语言 4秒

空间限制：C/C++ 262144K，其他语言 524288K

64bit IO Format: %lld

本题可使用本地IDE编码，不做跳出限制，编码后请点击“保存并调试”按钮进行代码提交。

■ 题目描述

小Q在做一个给人取名的程序。有 n 个名字，每个名字是一个字符串。陆续有 m 个人找你取名字，每个人有一个要求 $(str1, str2)$ ，他的名字要以 $str1$ 为前缀，但不能以 $str2$ 为前缀。你要在所有满足条件的名字中选字典序最小的给他。如果没有满足条件的名字就输出-1。一个名字如果用过了就不能再用。

输入描述:

第一行两个数 n, m 。

接下 n 行每行一个字符串，表示你初始有的名字，保证没有两个字符串完全一样。

接下 m 行每行两个字符串 $str1, str2$ 用空格隔开，表示一次询问。

输入字符串总长度 $\leq 300000, 1 \leq m \leq n \leq 100000$

输出描述:

共 m 行，每行一个字符串或'-1'表示答案

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
5 4
aaab
aab
aa
aabb
a
a c
aa aab
aaa aaab
a aaa
```

输出

复制

```
a
aa
-1
aab
```

```

#include <iostream>
#include<vector>
#include<algorithm>
#include<string>
using namespace std;

int In(string s1, string s2)
{
    int flag = 1;
    for (int i = 0; i < s1.size(); i++)
        if (s1[i] != s2[i])
            flag = 0;
    return flag;
}

int main()
{
    int m, n;
    cin >> n >> m;
    vector<string>a;
    string s, s1;
    for (int i = 0; i < n; i++)
    {
        cin >> s;
        a.push_back(s);
    }
    sort(a.begin(), a.end());
    vector<vector<string>>b(100000);
    for (int i = 0; i < m; i++)
    {
        cin >> s >> s1;
        b[i].push_back(s);
        b[i].push_back(s1);
    }
    string result;
    for (int i = 0; i < m; i++)
    {
        int flag = 0;
        for (int j = 0; j < a.size(); j++)
        {
            if (In(b[i][0], a[j]) && (!In(b[i][1], a[j])))
            {
                cout << a[j] << endl;
                a.erase(a.begin() + j);
                flag = 1;
                break;
            }
        }
        if (flag == 0)
            cout << "-1" << endl;
    }
}

```

