

## Оглавление

А. Ф. И. О.....	1
В. Шахматы.....	1
С. Последовательности .....	2
Д. Проверка памяти.....	3
Е. Наоборот.....	3
Ф. Постаматы.....	4

## А. Ф. И. О.

## В. Шахматы

Заметим, что полное решение включает в себя вычисление по формуле. Самое сложное – вычислить число клеток на диагонали. Необходимо помнить, что на диагоналях для всех клеток диагоналей либо сумма, либо разность координат одинаковая (такая же, как у клетки, на которой находится ферзь). Приведем решение на языке C++.

```
#include <bits/stdc++.h>

using ll = long long;
using namespace std;

int main(){
    ll x, y, m, n;
    ll ans = 0;

    cin >> m >> n >> x >> y;

    ans += n - 1;
    ans += m - 1;

    ans += min(n, m - x + y) - max(1ll, 1 - x + y) + 1;
    ans += min(n, x + y - 1) - max(1ll, x + y - m) + 1;

    cout << ans - 2;
}
```

## С. Последовательности

Приведем полное эффективное решение на языке C++, в котором реализуется идея динамического программирования, а именно подсчитываются последовательности, для которых количество нулей равно  $i$ , а количество единиц  $= j$ , то есть параметрами динамики будут количество нулей и единиц.

```
#include <bits/stdc++.h>

using LL = long long;

using namespace std;

LL w[30][30];

int n, K;

LL ans;

LL cnt(int i, int j){
    if (i + j == 0){
        w[i][j] = 1;
        return 1;
    }
    if (w[i][j] > 0)
        return w[i][j];
    if (i > 0) {
        w[i][j] += cnt(i - 1, j);
    }
    if (j > 0){
        w[i][j] += cnt(i, j - 1);
    }
    return w[i][j];
}

int main(){
    cin >> n >> K;
    for (int i = 0; i <= min(K, n); i++)
        for (int j = 0; j <= min(K, n); j++)
            if (i + j > 0 && i + j == n){
                ans += cnt(i, j);
            }
}
```

```
    }  
    cout << ans;  
}  
  
cout << ans;  
}
```

## D. Проверка памяти

Приведем полное решение на языке Python. Эффективное решение включает в себя использование структуры `set`. Кроме того, поскольку все числа не превосходят 10000, достаточно проверять делимость в цикле от 1 до квадратного корня из 10000, который равен 100.

```
n = int(input())  
se = set()  
res = 0  
a = list(map(int, input().split()))  
for i in range(n):  
    x = a[i]  
    for j in range(1, 101):  
        if (x % j == 0) and (j in se or x//j in se):  
            res += 1  
            break  
    se.add(x)  
print(res)
```

## E. Наоборот

Приведем полное решение на языке C++. Эффективное решение основано на методе сортировки подсчетом. Необходимо делать проверку строк на то, что они красивые, и пересчитывать в цикле число красивых строк, которые заканчиваются на каждую из букв латинского алфавита, чтобы потом вычислять, сколько красивых строк могут быть склеены с последующими красивыми строками.

```
#include <bits/stdc++.h>  
  
using namespace std;  
using ll = long long;  
int n;
```

```

ll ans = 0;
int st[26], f[26];

bool isGood(const string& s){
    char ol = 'a';
    for (auto c: s){
        if (ol > c)
            return false;
        ol = c;
    }
    return true;
}

int main(){
    cin >> n;
    string s;
    for (int i = 0; i < n; i++){
        cin >> s;
        char en = s.back();
        if (isGood(s))
        {
            for (int i = 0; i <= s[0] - 'a' ; i++)
                ans += f[i];
            f[en - 'a']++;
        }
    }
}

```

## F. Постаматы

Приведем полное решение на языке C++. Эффективное решение включает в себя использование структуры set.

```

#include <iostream>
#include <set>
using namespace std;

```

```

typedef long long ll;

int main(){
    ios::sync_with_stdio(0); // для ускорения ввода
    cin.tie(0);
    int n, a, q, k, x;
    cin >> n;
    set<pair<int, int>> se;
    for(int i = 1; i <= n; ++i){
        int a;
        cin >> a;
        se.insert({i, a});
    }
    cin >> q;

    for(int i = 0; i < q; ++i){
        ll d = 0;
        cin >> x >> k;
        auto it = se.lower_bound({x, 0});
        if(it == se.end()) {
            it = se.begin();
            x -= n;
        }
        while(k > 0){
            int f = it->first;
            int s = it->second;
            auto itr = it;
            itr++;
            se.erase(itr);
            if(s > k){
                d += (ll)k*(f-x);
                se.insert({f, s - k});
            }
        }
    }
}

```

```
    }  
    else  
    {  
        d += (ll)s*(f-x);  
        if(it == se.end()) {  
            it = se.begin();  
            x -= n;  
  
        }  
    }  
    k -= s;  
}  
cout << d << "\n";  
}  
}
```

## Оглавление

В. Карточки .....	1
С. Алфавитный справочник .....	2
D. Призы .....	3
Е. Спортивное программирование.....	3
Ф. Забор .....	4

## В. Карточки

У вас есть несколько карточек, на каждой карточке записана одна цифра. С использованием этих карточек вам нужно составить минимальное число, делящееся на 9. Каждую карточку можно использовать только один раз. Если какая-то цифра записана на нескольких карточках, то эту цифру в ответе можно использовать любое число раз, не превосходящее количества карточек, на которых она записана.

Найдите ответ для следующих возможных наборов карточек. Запишите ответ для каждого набора с новой строки в виде целого числа, то есть только цифрами без пробелов и запятых.

- а) На карточках записаны числа 1, 7, 5, 4, 6, 2.
- б) На карточках записаны числа 8, 3, 1, 2, 6, 5.
- в) На карточках записаны числа 1, 1, 1, 1, 2, 2, 3, 3, 3, 5.
- г) На карточках записаны числа 6, 6, 8, 8, 4, 4, 2, 2, 4, 6, 2.
- д) На карточках записаны числа 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4.

Во всех пунктах на карточках нет цифры 9, поэтому ответ нужно получить, выбрав минимальное число карточек, дающих в сумме 9.

- а) На карточках записаны числа 1, 7, 5, 4, 6, 2.

Ответ: 27

- б) На карточках записаны числа 8, 3, 1, 2, 6, 5.

Ответ: 18

- в) На карточках записаны числа 1, 1, 1, 1, 2, 2, 3, 3, 3, 5.

Ответ: 135

- г) На карточках записаны числа 6, 6, 8, 8, 4, 4, 2, 2, 4, 6, 2

Ответ: нельзя подобрать карточки, чтобы они в сумме давали 9. Попробуем составить число 18. Это можно сделать, используя 3 карточки. Ответ: 288.

д) На карточках записаны числа 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4.

Ответ:

Попробуем перебрать, например, количество цифр 2 в записи ответа. Если не использовать 2 совсем, а использовать только цифры 4, то чтобы составить кратное 9, понадобится девять цифр 4.

Также переберем число цифр 2:

- 24444
- 2244444444
- 222444
- 2222444444
- 2222244
- 222222444444

Ответ: 24444.

### С. Алфавитный справочник

Вам предлагается 4 «слова» на английском языке. Для каждого из этих «слов» предложите слово, которое можно получить из этого слова перестановкой каких-то двух не обязательно соседних букв и которое при этом является минимально возможным в лексикографическом (алфавитном) порядке. Например, для слова «АВВА» ответом будет «ААВВ». Напомним, что из двух слов в лексикографическом порядке одно будет меньше другого, если у этих слов есть какая-то общая совпадающая начальная часть (возможно, пустая), а следующий символ одного слова идёт в алфавите раньше, чем следующий символ другого слова.

Запишите ответы для каждого из четырёх «слов» ниже, каждый ответ — в отдельной строке. В каждой строке должны быть только английские буквы, номер задания указывать не нужно. В вашем ответе должно быть ровно 4 строки. Если вы не можете дать ответ на какое-то задание, напишите любую строку из прописных английских букв, например, «А».

Приведем английский алфавит:

ABCDEFGHIJKLMNOPQRSTUVWXYZ.

Вот эти слова:

1. DGZYGX
2. QAYQQAFOCAD
3. BCJKMJNJOJPYKY
4. DEFGHMLNQRТУVWZYX

ОТВЕТ:

1. DGGYZX
2. AAYQQAFOCQD

3. BCJJMJNJOKPYKY
4. DEFGHLMNORTUVWZYX

## D. Призы

Приведем решение на языке C++.

```
#include <bits/stdc++.h>
using namespace std;
int main(){
int n, a;
int ans;
cin >> n >> a;
cout << a + (n - a % n) % n;
}
```

## E. Спортивное программирование

Приведем решение на языке Python.

```
n = int(input())
d, L, c = map(int, input().split())
ans = 0
early = d
for i in range(1, n):
    a, b, t = d, L, c
    d, L, c = map(int, input().split())
    if(d + L <= a + b and d >= a and c != t):
        ans = d
        break
print(ans - early)
```

## F. Забор

Приведем решение на языке Python.

```
L = int(input())
N = int(input())
ans = 0
last_start = -10 ** 6
for i in range(N):
    fence_elem = int(input())
    if fence_elem == 1 and i >= last_start + L:
        ans += 1
        last_start = i
print(ans)
```