▷ Задача 1. Спартанцы, в атаку!

Царь Леонид решил повести своё войско в наступательный поход.

Он построил воинов в один ряд, оглядел их и решил проверить их боеспособность.

Для проверки боеспособности он придумал следующую схему:

Он подошёл к некоторым бойцам, и шепнул им на ухо команду "в бой!".

За одну секунду боец, который знает команду, может передать её правому и левому соседу в ряду.

Леонид хочет узнать, за сколько секунд все бойцы войска узнают команду?

Воины представлены бинарной строкой. (1 - если воин изначально знал команду, 0 - если не знал)

Входные данные

На ввод подаётся строка из нулей и единиц. Длина строки не превосходит 10^5.

Выходные данные:

В ответе выведите единственное число - минимальное количество секунд, через которое все спартанцы уже будут знать команду.

Ограничение времени выполнения программы:

1 секунда

Примеры тестовых данных:

Входные данные	Результат работы программы
10001	2
1000101	2
00000000000000000000000000000000100000 00001000000000000000000000000000000 00000000000000000000000000	58

Решение задачи на языке Python

```
a = input()
s = a.split('1')
q0 = len(s[0])
q1 = 0
q = q0
for i in range(len(s)-2):
    x = len(s[i+1])/2
    if x==int(x):
        q3 = int(x)
    else:
        q3 = int(x)+1
    if q1<q3:
        q1=q3
q2 = len(s[-1])
if q0 < q1:
    q = q1
if q1 < q2:
    q = q2
print(q)
main(k)
```

▷ Задача 2. Не короткое замыкание

Вам дана парта размера $N \times M$ — клетчатое поле. В каждой клетке либо находится электрический элемент (он обозначается за #), либо ничего не находится (*).

Два элемента соединены, если они находятся в клетках, граничащих по стороне.

Гарантируется, что цепь связная и незамкнутая.

Вову заинтересовал вопрос: если замкнуть какие-то два элемента (например, соединить их проволокой), то сколько в замыкании максимум будет участвовать элементов?

(Примеч: элемент находится в замыкании, если он находится в цикле, который образовался, когда Вова замкнул два элемента).

Входные данные

В первой строке записано два целых числа: N и $M (2 \leqslant N, M \leqslant 100)$ — высота и ширина парты соответственно. После этого идут N строк длины M — описание парты.

Выходные данные:

Выведите максимальную длину цикла, который получится, если замкнуть какие-то два элемента.

Ограничение времени выполнения программы:

1 секунда

Примеры тестовых данных:

Входные данные	Результат работы программы
3 3 ### ..# .##	6
4 5 ..#.. .##.. ..##.	4
5 4 .#.. ##.. .### ##.# .#..	6

Решение на C++

С заданием полностью справились только два участника. Приводим решение призера отборочного тура, Адиля Канатбекова, 9 класс, г. Бишкек.

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long
#define pb push_back
#define db double
#define pll pair<ll, ll>
```

```
#define fs first
#define sc second
ll n,m;
vector<ll> dx{1, 0, -1, 0};
vector<ll> dy{0, 1, 0, -1};
vector<vector<char>> g;
vector<vector<ll>> dist1, dist2;
pll dfs(pll a, vector<vector<ll>> &dist){
    pll mx={0, 0};
    ll x=a.fs;
    ll y=a.sc;
    for(ll i=0; i<4; i++){
        if(x+dx[i]>0 && x+dx[i]<=m && y+dy[i]>0 && y+dy[i]<=n && g[x+dx[i]][y+dy[i]]=='#'
        && dist[x+dx[i]][y+dy[i]]==-1){
            dist[x+dx[i]][y+dy[i]]=1;
            dist[x+dx[i]][y+dy[i]]=dist[a.fs][a.sc]+1;
            if(dist[mx.fs][mx.sc]<dist[x+dx[i]][y+dy[i]]) mx={x+dx[i], y+dy[i]};
            pll mx1=dfs({x+dx[i], y+dy[i]}, dist);
            if(dist[mx1.fs][mx1.sc]>dist[mx.fs][mx.sc]) mx=mx1;
        }
    }
    return mx;
}
int main(){
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    cin>>n>>m;
    pll id;
    g.resize(m+1, vector<char>(n+1));
    dist1.resize(m+1, vector<ll>(n+1, -1));
    dist2.resize(m+1, vector<ll>(n+1, -1));
    for(ll i=1; i<=n; i++){
        for(ll j=1; j<=m; j++){
            cin>>g[j][i];
            if(g[j][i]=='#') id={j, i};
        }
    }
    dist1[id.fs][id.sc]=1;
    pll mx=dfs(id, dist1);
    dist2[mx.fs][mx.sc]=1;
    pll ans=dfs(mx, dist2);
    cout<<dist2[ans.fs][ans.sc]<<"\n";
}
```

▷ Задача 3. Публикан

«Публикан», или иначе говоря, «сборщик налогов» обладал одной из самых популярных профессий в Древнем Риме. В то время деньги собирались с граждан вручную, а периодические территориальные войны лишь добавляли неразберихи в общую финансовую суматоху.

В суматохе публикан Марк не может сосредоточиться и сложить два римских числа, чтобы посчитать суммарный налог, который ему нужно взыскать у супружеской пары. Пожалуйста, помогите ему это сделать.

Входные данные

В двух строках содержится по одному числу, записанному в римской системе счисления (каждое число не превосходит 3999).

Выходные данные:

В ответе запишите число, являющееся суммой двух данных римских чисел (так же в римской системе счисления).

Ограничение времени выполнения программы:

1 секунда

Примеры тестовых данных:

Входные данные	Результат работы программы
I I	II
IV IX	XIII
D XXIV	DXXIV

Решение задачи на языке Python

```
all_roman = [(1000, 'M'), (900, 'CM'), (500, 'D'), (400, 'CD'), (100, 'C'),
(90, 'XC'), (50, 'L'), (40, 'XL'), (10, 'X'), (9, 'IX'), (5, 'V'), (4, 'IV'),
 (1, 'I')]

def t(num):
    roman = ''
    while num > 0:
        for i, r in all_roman:
            while num >= i:
                roman += r
                num -= i
    return roman
def v(rom):
    dec = 0
    for i, r in all_roman:
        while rom.startswith(r):
            dec += i
            rom = rom[len(r):]
    return dec

s1=input()
s2=input()

a1=v(s1)
a2=v(s2)

a=a1+a2

s=t(a)

print(s)
```

▷ **Задача 4. Чебурашка и апельсины**

Чебурашка очень любит апельсины. В магазине, где он живёт, продавец иногда даёт Чебурашке его любимое лакомство.

Однажды когда Чебурашке стало скучно, он решил взять все свои апельсины и положить их на чашечные весы так, чтобы весы встали в положение равновесия.

Помогите Чебурашке - сообщите ему, может ли он положить свои апельсины на весы нужным образом.

Входные данные

Первая строка содержит целое положительное число $N(2 \leqslant N \leqslant 20)$ — количество апельсинов.

Вторая строка содержит N натуральных чисел — веса апельсинов, которые не превосходят 10^6.

Выходные данные:

Выведите единственную строку:

NO - если нельзя положить апельсины на весы

YES - если можно расположить апельсины на весах требуемым образом

Ограничение времени выполнения программы:

1 секунда

Примеры тестовых данных:

Входные данные	Результат работы программы
4 1 2 3 4	YES
10 560969 173071 784074 994974 581388 391519 76845 939307 420796 59099	YES
10 35762 258056 461849 499400 977649 613476 430517 825321 23036 56093	NO

Решение задачи на языке Python

```
def can_balance_oranges(N, weights):
    total_weight = sum(weights)

    for i in range(1 << N):
        current_weight = 0

        for j in range(N):
            if i & (1 << j):
                current_weight += weights[j]

        if current_weight == total_weight / 2:
            return "YES"

    return "NO"


N = int(input())
weights = list(map(int, input().split()))

print(can_balance_oranges(N, weights))
```

▷ **Задача 5. Важное сообщение**

В Солнечный город по радиосвязи поступило сообщение с Луны. В связи с тем, что при передачи сообщения на такое большое расстояние могут быть помехи, коротышки с Луны воспользовались шифром с защитой от повреждения данных. Каждый бит сообщения повторяется три раза перед отправкой. Например, строка "101"преобразуется к виду "111000111".

Весь процесс кодирования выглядит следующим образом:

1. Символ заменяется своим ASCII кодом
2. ASCII код конвертируется двоичную систему счисления (8 бит на символ)
3. каждый полученный бит утраивается.

Известно, что в каждой "тройке"битов не может быть больше одной ошибки и что не может прийти два бита вместо трех. Т.е. возможна ситуация, когда вместо 000 пришло 001 и невозможна, когда пришло 00.

Помогите Знайке написать программу для расшифровки сообщения

Входные данные:

Строка из нулей и единиц, длиной не более 1000 символов – зашифрованное послание

Выходные данные:

Расшифрованное сообщение

Ограничение времени выполнения программы:

1 секунда

Примеры тестовых данных:

Входные данные	Результат работы программы
000111111000000000000111	a
000111111000000000111000	b
000111111000000000000111000111111 000000000111000000111111000000000 111111	abc

Решение задачи на языке Python

```
def decode(binary):
    reduced = (get_digit(triplet) for triplet in chunks(binary, 3))
    return "".join(get_char(byte) for byte in chunks("".join(reduced), 8))


def chunks(seq, size):
    return (seq[i:i+size] for i in range(0, len(seq), size))

def get_digit(triplet):
    return max(triplet, key=triplet.count)


def get_char(byte):
    return chr(int(byte, 2))

print(decode(input()))
```

▷ **Задача 6. Проклятый многоугольник**

Капитан Джек Воробей и его команда путешествуют на Черной Жемчужине по Карибскому морю. Чтобы найти корабль-призрак, им нужно попасть в проклятый многоугольник.

Известна текущая координата корабля и координаты вершин многоугольника. Помогите Джеку определить, на месте ли они.

Входные данные:

Координата Черной Жемчужины x, y.

Координаты точек проклятого многоугольника

Выходные данные:

YES – если корабль в многоугольнике

NO – если нет

Ограничение времени выполнения программы:

1 секунда

Примеры тестовых данных:

Входные данные работы программы	Результат
1 1 -4 -4 4 -4 4 4 -4 4	YES
3.9 0 -4 -4 4 -4 4 4 -4 4	YES
6 1 -4 -4 4 -4 4 4 -4 4	NO

Решение задачи на языке Python

С этой задачей полностью справились только три участника отборочного тура. Приводим решение призера отборочного тура Матвея Непеина, 9 класс, г. Ноябрьск

```
def is_inside_polygon(x, y, polygon):
    n = len(polygon)
    intersections = 0
    epsilon = 1e-9
    for i in range(n):
        x1, y1 = polygon[i]
        x2, y2 = polygon[(i + 1) % n]
        if (y1 > y) != (y2 > y) and x < (x2 - x1) * (y - y1) / (y2 - y1) + x1 \
        + epsilon:
            intersections += 1

    return intersections % 2 == 1
ship_x, ship_y = map(float, input().split())
polygon = []
while True:
```

```
    try:
        x, y = map(float, input().split())
        polygon.append((x, y))
    except EOFError:
        break
if is_inside_polygon(ship_x, ship_y, polygon):
    print("YES")
else:
    print("NO")
```