МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Задача: Поиск сокровищ в лабиринте

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

У вас есть карта лабиринта в виде матрицы размером N·M, где каждая ячейка может быть свободной (обозначается как '.') или стеной (обозначается как '#'). В одной из свободных ячеек находится сокровище (обозначено как 'S'), а в другой - вход в лабиринт (обозначен как 'E'). Ваша задача - написать программу, которая найдёт кратчайший путь от входа до сокровища, перемещаясь только по свободным ячейкам. Путь можно проложить вверх, вниз, влево или вправо, но нельзя двигаться по диагонали.

Формат ввода

Первая строка ввода содержит два целых числа N и M ($1 \le N, M \le 100$), разделенных пробелом, обозначающих размеры лабиринта.
Последующие N строк содержат по M символов, описывающих лабиринт.

Формат вывода

Программа должна вывести одно целое число - количество шагов в кратчайшем пути от входа до сокровища. Если путь найти невозможно, программа должна вывести -1.

Пример

Ввод	Вывод
1 5 E.S# #	2

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Пример решения

```
def is_valid_move(cells, row, col, visited):
    if row >= 0 and row < len(cells) and col >= 0 and col < len(cells[0]):
        if cells[row][col] != '#' and not visited[row][col]:
            return True
    return False


def find_shortest_path(cells, start_row, start_col, end_row, end_col):
    visited = [[False for _ in range(len(cells[0]))] for _ in range(len(cells))]
    queue = []
    queue.append((start_row, start_col, 0))
    visited[start_row][start_col] = True

    while queue:
        current_row, current_col, distance = queue.pop(0)


        if current_row == end_row and current_col == end_col:
            return distance


        for row_offset, col_offset in [(-1, 0), (1, 0), (0, -1), (0, 1)]:
            new_row = current_row + row_offset
            new_col = current_col + col_offset

         if is_valid_move(cells, new_row, new_col, visited):
                queue.append((new_row, new_col, distance + 1))
                visited[new_row][new_col] = True

    return -1
N, M = map(int, input().split())

maze = []
for _ in range(N):
    row = input().strip()
    maze.append(list(row))

start_row, start_col = None, None
end_row, end_col = None, None
for row in range(N):
```

```
    for col in range(M):
        if maze[row][col] == 'E':
            start_row, start_col = row, col
        if maze[row][col] == 'S':
            end_row, end_col = row, col

shortest_path = find_shortest_path(maze, start_row, start_col, end_row, end_col)

print(shortest_path)
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Задача: Робот перемещается по четырем точкам

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Перед роботом стоит задача поиска самого короткого маршрута на основании заданных точек. Каждый маршрут формируется из четырех точек.

Робот перемещается по трем участкам: равноускоренному (точка №1 - точка №2 со ускорением $a1=2м/с^2$), равномерному (точка №2 - точка №3) с ускорением $a2=0м/с^2$) и равнозамедленному (точка №3 - точка №4 с ускорением $a3=-1с2м$).

Начальная скорость робота принимается равной 0м/с, при равномерном движении скорость равна 4м/с.

Роботу необходимо рассчитать длину самого короткого пути и суммарное время прохождения самого короткого пути.

Формат ввода

На вход программы поступают 6 строк, каждая из которых состоит из двух вещественных чисел, разделенных разделителем запятой и пробелом (", ").

Первое число представляет собой координату X точки маршрута, второе число координату Y точки маршрута: X, Y.

X и Y измеряются в метрах.

X принимает значения из диапазона $\in[0;5]\in[0;5]$ метров, Y принимает значения из диапазона $\in[0;5]\in[0;5]$ метров.

Формат вывода

На выходе программное обеспечение должно выдавать рассчитанные значения длины самого короткого маршрута и значение времени прохождения самого короткого маршрута.

Значение длины самого короткого маршрута записывается в первую строку. Значение указывается в метрах. Значение необходимо округлить до двух знаков после запятой.

Значение времени прохождения самого короткого маршрута записывается во вторую строку. Значение указывается в секундах. Значение необходимо округлить до двух знаков после запятой.

Пример 1

Ввод	Вывод
1.49, 4.53 4.26, 4.69 3.02, 1.53 0.43, 2.32 4.9, 4.87 4.61, 2.86	4.60 1.77

Пример 2

Ввод	Вывод
4.54, 4.65 2.05, 1.18 2.15, 1.69 3.55, 4.72 2.84, 0.36 3.55, 4.29	3.90 1.52

Пример 3

Ввод	Вывод
3.12, 2.48 1.51, 3.29	4.04 1.74

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Ввод	**Вывод**
3.46, 4.55 2.21, 2.53 1.03, 0.96 4.24, 0.55	

Пример решения

```
from itertools import permutations
from math import sqrt, dist

points = list([tuple(map(float, input().split(', '))) for i in range(6)])
variants = list(permutations(points, 4))
results = []

for element in variants:
    dist_1 = dist(element[0], element[1])
    dist_2 = dist(element[1], element[2])
    dist_3 = dist(element[2], element[3])

    t_1 = sqrt(dist_1)
    t_2 = dist_2 / 4
    t_3_1 = -4 + (16 + 2 * dist_3) ** 0.5
    t_3_2 = -4 - (16 + 2 * dist_3) ** 0.5

    if t_3_1 >= 0 >= t_3_2:
        t_3 = t_3_1
    elif t_3_2 >= 0 >= t_3_1:
        t_3 = t_3_2
    else:
        if 4 - t_3_1 < 0:
            t_3 = t_3_2
        elif 4 - t_3_2 < 0:
            t_3 = t_3_1
        else:
            t_3 = min(t_3_1, t_3_2)

    results.append((dist_1 + dist_2 + dist_3, t_1 + t_2 + t_3))
```

```
answer = min(results, key=lambda x: x[1])

print((str(abs(round(answer[0], 2))) + '0000')[:4])
print((str(abs(round(answer[1], 2))) + '0000')[:4])
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Задача: Проксима Центавра

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

До ближайшей к нам звезды, Проксима Центавра, примерно 4,244 световых лет. Современные химические реактивные двигатели с трудом развивают скорость до 20 км/с, что делает невозможным достигнуть Проксима Центавра за разумное время.

В 1960 году физик Роберт Буссард предложил идею прямоточного ядерного двигателя, который использует в качестве рабочего тела водород – основной компонент в составе межзвёздной среды. Корабль с таким двигателем мог бы собирать себе топливо во время движения из окружающего пространства, затем водород нагревали бы до нескольких миллионов градусов для термоядерного синтеза, из чего извлекали бы энергию для движения. Такой двигатель при весе около 1000 т, может в теории поддерживать постоянное ускорение равное $g=9{,}81 м/с^2$.

Допустим, ракета массой M т (с учетом полезной нагрузки) и прямоточным ядерным двигателем после выхода на околоземную орбиту начинает разгон с постоянным ускорением kg. Через T0 недель разгон прекращается и далее поддерживается постоянная скорость. Через t лет скорость начинают снижать за n этапов по следующему плану: в течение T1 недель скорость гасят с ускорением kg/2, затем в течение T2 недель – с ускорением kg/4, далее в течение T3 недель – с ускорением kg/8 и так далее. В конце описанных событий ракета останавливается около новой планеты на расстоянии R св. лет от Земли.

Определить пройденное до планеты расстояние R (св. лет).

Использовать скорость света $c=3\cdot 108 м/с$. Считать, что в году 365 дней. Решать задачу с использованием формул классической механики (классических формул Галилея).

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Формат ввода

На вход подается 4+n вещественных числа n,M,t,k,T_0,T_1,T_2,...,T_n, каждое число подаётся с новой строки.

Формат вывода

Вывести вещественное число R, округлённое до сотых, которое представляет собой расстояние от Земли до новой планеты в световых годах.

Пример

Ввод	Вывод
2 202 4 40 0.5 5 10	1.99

Пример решения

```
n = int(input())
M = float(input())
t = float(input()) * 365 * 24 * 3600
k = float(input())
g = 9.81
T0 = 0
T = [0] * (n + 1)
for i in range(1, n + 1):
    T[i] = float(input()) * 7 * 24 * 3600
    T0 += T[i] / 2 ** i
R0 = k * g * T0 ** 2 / 2
r = k * g * T0 * t
R = [0] * (n + 1)
R[0] = R0
P = k * g * T0
for i in range(1, n + 1):
    Tdiff = 0
```

```
    if i == 1:
        Tdiff -= k * g * T[n] / 2 ** (n + 1)
    else:
        for m in range(1, i):
            Tdiff -= k * g * T[m] / 2 ** m
        Tdiff -= k * g * T[i] / 2 ** (i + 1)
    R[i] = T[i] * (P + Tdiff)
R_ = (r + sum(R)) / (3e8 * 365 * 24 * 3600)
print(round(R_, 2))
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Задача: Рамка в магнитном поле

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Квадратная рамка, изготовленная из проводящего материала, имеет сторону размера **a** (см). Рамка находится в магнитном поле с индукцией B=0,5Тл.

Через середины двух противоположных сторон рамки проходит ось, вокруг которой рамку вращают с частотой **n** ($с^{-1}$).

Линии вектора магнитной индукции направлены перпендикулярно оси вращения. Вычислите мгновенное значение ЭДС индукции (в вольтах), возникающей в рамке с точностью до трёх знаков после запятой в промежуток времени $t \in [0;5,6]$ с шагом 0,1 секунды.

В начальный момент времени магнитный поток через рамку максимальный.

Формат ввода

На вход подаются данные в виде строки из двух чисел, разделенных пробелом, где первое число – сторона квадратной рамки a в см, второе – частота вращения рамки вокруг оси n в $с^{-1}$.

Формат вывода

Необходимо вывести медианное значение ЭДС за заданный промежуток времени (в вольтах).

Ответ выводить с округлением до трех знаков после запятой.

Пример

Ввод	Вывод
10 1.35	0.004

Примечания

Если выходным значением является целое число, его необходимо выводить с дробной частью, равной нулю и одним знаком в дробной части.

Разделителем является точка.

В случае получения
значений: 1.001.00, 11, 1.0001.000, 1.000000000000011.0000000000001,
выводить следует: 1.01.0.

Пример решения

```
import math

a, n = [float(x) for x in input().split()]
a = a / 100
S = a * a
B = 0.5
t = 0
data = []
while t != 5.7:
    e = 2 * math.pi * n * B * S * math.sin(2 * math.pi * n * t)
    data.append(round(e, 3))
    t = round(t + 0.1, 2)
data.sort()
print(str(data[len(data) // 2]).ljust(5, '0'))
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Задача: Терминатор и загадка математики

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В альтернативной вселенной, где Терминатор решил сделать перерыв в своих обычных действиях, направленных против человечества, он обнаружил страсть к математике. Вместо того чтобы спасать мир, теперь он хочет решать уравнения. У него есть для вас особое задание.

Терминатор предлагает уравнение $2x \cdot \sqrt{1-x^2} = a \cdot (1-2x^2)$ и просит найти все его решения в интервале $x \in (-1,1)$ для заданного коэффициента a. Решения должны быть точными до четырех знаков после запятой.

Формат ввода

Первая строка содержит целое число T, количество тестовых случаев. Каждая из следующих T строк содержит одно число с плавающей точкой a, коэффициент для тестового случая.

Формат вывода

Выведите строку, начинающуюся с "Case #x:", где x - номер тестового случая (начиная с 1). После этого выведите все решения для этого тестового случая, каждое округленное до четырех знаков после запятой. Если решения не существует, напечатайте "Нет решения". Каждое решение должно быть разделено пробелом. Если существует несколько решений, упорядочите их от наименьшего к наибольшему.

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
Заключительный этап
Продуктовый сектор
Междисциплинарные задачи
11 класс

Пример

Ввод	Вывод
2 -1.6572578577847517 -1.1272783495542056	Case #1: -0.4916 0.8708 Case #2: -0.4101 0.9120

Пример решения

```
import math

def find_solutions(a):
    sol1 = 0.5 * math.sqrt(2 - 2 * math.sqrt(1 - a**2 / (a**2 + 1)))
    sol2 = -sol1
    sol3 = 0.5 * math.sqrt(2 + 2 * math.sqrt(1 - a**2 / (a**2 + 1)))
    sol4 = -sol3
    if a <= 0:
        return [sol2, sol3]
    else:
        return [sol4, sol1]

def main():
    T = int(input().strip())
    results = []
    for i in range(1, T + 1):
        a = float(input().strip())
        solutions = find_solutions(a)
        if solutions:
            results.append(f"Case #{i}: {' '.join(f'{sol:.4f}' for sol in solutions)}")
        else:
            results.append(f"Case #{i}: No Solution")

        print(*results, sep=" ")

if __name__ == "__main__":
    main()
```