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

Задача: Острова Сокровищ

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

Архипелаг состоит из N островов ($1 \le N \le 100$), некоторые из которых соединены мостами. Каждый мост соединяет ровно два различных острова, и по каждому мосту можно перемещаться в обе стороны. На каждом острове спрятаны сокровища, стоимость которых известна. Цель - собрать сокровища настолько, насколько это возможно, но при этом можно посетить не более K островов ($1 \le K \le N$). Вы начинаете путешествие с любого острова. Ваша задача - написать программу, которая определит максимальную стоимость сокровищ, которые можно собрать, не посещая более K островов.

Формат ввода

Первая строка содержит два целых числа N и K. Вторая строка содержит N целых чисел, обозначающих стоимость сокровищ на каждом острове. Каждая из следующих $N - 1$ строк содержит пару чисел, обозначающих острова, соединённые мостом.

Формат вывода

Программа должна вывести одно целое число - максимальную стоимость сокровищ, которую можно собрать, посетив не более K островов.

Пример

Ввод	Вывод
5 3 10 40 30 50 20 1 2 2 3	120

Ввод	Вывод
3 4 4 5	

Примечания

В примере, начиная с первого острова, вы можете посетить острова 1 → 2 → 3, собрав сокровища на общую сумму 10 + 40 + 30 = 80, или острова 2 → 3 → 4, собрав сокровища на общую сумму 40 + 30 + 50 = 120, что является оптимальным решением.

Пример решения

```
def max_treasure(N, K, treasures, bridges):
    from collections import defaultdict, deque
    graph = defaultdict(list)
    for a, b in bridges:
        graph[a].append(b)
        graph[b].append(a)
    def bfs(start):
        visited = set()
        queue = deque([start])
        all_nodes = []
        while queue:
            node = queue.popleft()
            if node not in visited:
                visited.add(node)
                all_nodes.append(node)
                for neighbor in graph[node]:
                    if neighbor not in visited:
                        queue.append(neighbor)
        return all_nodes
    visited = set()
    max_value = 0
    for i in range(1, N + 1):
        if i not in visited:
            component_nodes = bfs(i)
            visited.update(component_nodes)
            if len(component_nodes) <= K:
                max_value += sum(treasures[node - 1] for node in component_nodes)
```

```
        else:
            max_value += sum(sorted((treasures[node - 1] for node in
component_nodes), reverse=True)[:K])
    return max_value

N, K = map(int, input().split())
treasures = list(map(int, input().split()))
bridges = []
for _ in range(N-1):
    i1, i2 = map(int, input().split())
    bridges.append((i1, i2))

print(max_treasure(N, K, treasures, bridges))
```

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

Задача: Самый короткий путь

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 координат заданных точек. Каждый маршрут формируется из четырех точек.

Роботу необходимо найти координаты точек, составляющих самый короткий маршрут, и рассчитать длину самого короткого пути.

Формат ввода

На вход программы поступают 6 строк, каждая из которых состоит из двух вещественных чисел, разделенных 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 $\in[0;5]$ метров, Y принимает значения из диапазона $\in[0;5]$ метров.

Формат вывода

На выходе программное обеспечение должно выдавать координаты точек самого короткого маршрута и рассчитанное значение длины самого короткого маршрута.

Значения координат точек самого короткого маршрута записываются последовательно в первых четырех строках.

Значения координат X и Y округляются до второй точки после запятой и разделяются пробелом.

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

Пример 1

Ввод	Вывод
0.00 0.00 2.00 3.00 3.00 2.00 4.00 5.00 1.00 4.00 5.00 1.00	1.00 4.00 2.00 3.00 3.00 2.00 5.00 1.00 5.06

Пример 2

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	3.02 1.53 4.61 2.86 4.26 4.69 4.90 4.87 4.60

Пример 3

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	2.05 1.18 2.15 1.69 3.55 4.29 3.55 4.72 3.90

Пример решения

```
import itertools
import math

def distance(point1, point2):
  return math.sqrt((point1[0] - point2[0])**2 + (point1[1] - point2[1])**2)

with open("input.txt", "r") as file:
points = [tuple(map(float, line.split())) for line in file]
```

```
combinations = itertools.permutations(points, 4)

shortest_route = None
shortest_distance = float('inf')

for route in combinations:

  route_distance = sum(distance(route[i], route[i+1]) for i in range(3))
  if route_distance < shortest_distance:
    shortest_distance = route_distance
    shortest_route = route

with open("output.txt", "w") as file:
  for point in shortest_route:
    file.write("{:.2f} {:.2f}\n".format(point[0], point[1]))
    file.write("{:.2f}".format(shortest_distance))
```

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

Задача: Проксима Центавра 10

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 1946 году Станислав Улам высказал идею космического корабля, использующего в качестве движителя серию ядерных мини-бомб, которые следовало выбрасывать позади ракеты, чтобы она двигалась, «седлая» ударную волну от взрывов. В конце 70-х гг. концепция звездолёта с ядерным движителем возродилась в проекте «Дедал» Британского межпланетного общества. Ракетный корабль по проекту «Дедал» оказался таким громадным, что строить его пришлось бы в открытом космосе. Он должен был весить 54 000 т (почти весь вес — ракетное топливо) и смог бы разгоняться до 7,1% скорости света, неся на себе полезную нагрузку весом 450 т.

Допустим, ракета с импульсным ядерным двигателем после выхода на околоземную орбиту начинает разгон с постоянным ускорением a_1, составляющим половину предельно возможного ускорения. Через T недель разгон прекращается и далее поддерживается постоянная скорость – p % от скорости света. Через некоторое время ракету планировали затормозить с тем же ускорением, чтобы она остановилась около новой планеты на расстоянии R св. лет от Земли. Однако из-за технического сбоя точку пространства, в которой следовало начать торможение, пропустили, обнаружив это лишь через τ дней. Теперь необходимо внести коррективы в полётный план, немедленно начав торможение с увеличенным ускорением a_2. Выясните необходимое ускорение и определите, может ли двигатель его обеспечить.

Определить необходимое ускорение при торможении a_2 ($м/с^2$). Возможно ли его достичь? Если необходимое для торможения в указанной точке ускорение достижимо, вывести «yes» и значение этого ускорение, если оно не достижимо, вывести «no» и расстояние x (10^9 км), на которое ракета удалится от цели, тут же начав торможение с максимально возможным ускорением.

Использовать скорость света $c = 3 \cdot 10^8$ м/с.

Считать, что в году 365 дней.

Формат ввода

На вход подается четыре вещественных числа, T , p , R , τ каждое число подаётся с новой строки.

Формат вывода

Вывести строку, содержащую вердикт "yes" или "no", и вещественное число, a_2 или x, округленное до сотых через пробел.

Пример

Ввод	Вывод
7 7.5 4.442 10	yes 8.98

Пример решения

```
T = float(input())
p = float(input())
R = float(input())
tau = float(input())
if tau > T * 7 / 4:
        print('no', round((2 * tau - T * 7) * 24 * p * 3e8 * 3.6e3 / (100 * 2 * 1e12), 2))
else:
        print('yes', round(p * 3e6 / ((T * 7 - 2 * tau) * 3.6e3 * 24), 2))
```

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

Задача: Увлажнитель воздуха

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

Фирма по производству увлажнителей воздуха заказала вам написать программу для производимых увлажнителей.

Дано: начальное значение относительной влажности воздуха в комнате (в процентах), требуемое значение относительной влажности воздуха в комнате (в процентах), объём комнаты (в $м^3$) и скорость распыления воды увлажнителем (г/мин).

Необходимо найти минимальное время (в минутах), за которое увлажнитель увеличит влажность в комнате с начальной до конечной.

Температуру в комнате считать равной 20℃.

Давление насыщенных паров при комнатной температуре – 2.34 кПа.

Молярная масса воды – 18 г/моль.

Универсальная газовая постоянная – 8.31 Дж / моль $*$ К.

При расчётах ответ округляется до минут в большую сторону.

Формат ввода

На вход подается четыре вещественных числа, разделенных пробелами: начальное значение относительной влажности воздуха в комнате (в процентах), требуемое значение относительной влажности воздуха в комнате (в процентах), объём комнаты (в м3) и скорость распыления воды увлажнителем (г/мин).

Формат вывода

Необходимо вывести минимальное время (в минутах), за которое увлажнитель увеличит влажность в комнате с начальной до конечной.

Пример

Ввод	Вывод
20 80 30 10 32	

Решение

```
pi0, pi1, V, s = [float(x) for x in input().split()]
s = s / 1000
RT = 8.31 * 293
k = 0.018 * 2340 * V / RT
result = (pi1 - pi0) * k / 100
t = 0
count = 0
while t < result:
    t += s
    count += 1
print(count)
```

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

Задача: Зачарованный лес

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

В далекой стране находится волшебный лес, охраняемый древним заклинанием. Граница этого заклинания определяется уникальными мистическими силами. По преданиям старейшин, форма заклинания символизируется через священные земные контуры: она ограничена кругом, центром которого служит Великое Дерево, а форму ему придают духи двух древних квадратов. Эти квадраты, известные как Квадраты Старейшин, были трансформированы временем, но их духи все еще присутствуют, переплетаясь вокруг круга.

Круг, называемый Охраняющим Кругом, располагает своим центром у Великого Дерева, координаты которого обозначаются как (cx, cy), и имеет радиус r. Центры обоих древних квадратов, преобразованных силами природы, точно совпадают с центром Охраняющего Круга. Диагонали этих квадратов, одного большего и одного меньшего, простираются на длины d1 и d2 соответственно.

Задачей является вычисление площади, заключенной между границами этих двух квадратов внутри Охраняющего Круга.

Формат ввода

На вход подаются данные в виде строки из пяти чисел, разделенных пробелом.

- cx и cy ($-10^5 \le cx, cy \le 10^5$): координаты центра Великого Дерева, совпадающие с центром Охраняющего Круга и центрами обоих квадратов.
- dr ($1 \le r \le 10^5$): радиус Охраняющего Круга.
- d1 и d2 ($1 \le d1, d2 \le 10^5$): длины диагоналей квадратов, где d1 — диагональ большего квадрата, а d2 — диагональ меньшего квадрата.

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

Формат вывода

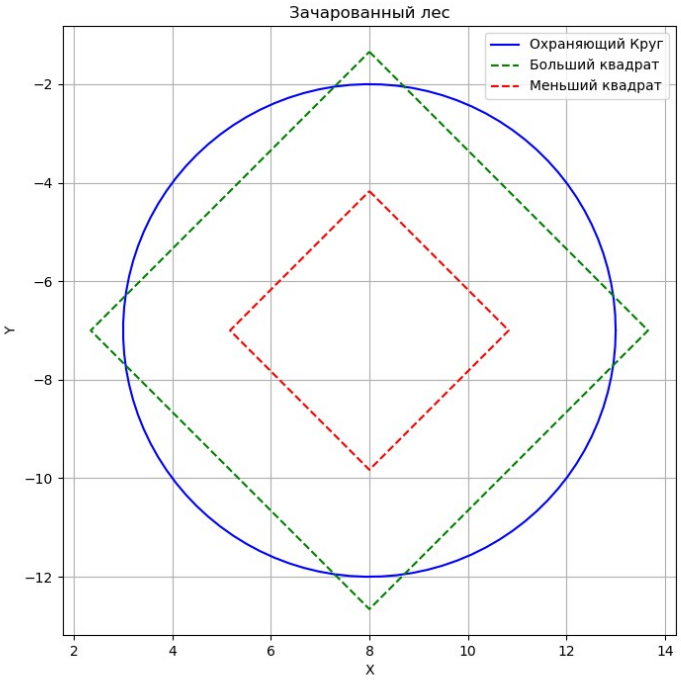
Выведите одно число — площадь волшебного леса, заключенную между двумя квадратами внутри Охраняющего Круга, с точностью до 6 знаков после запятой.

Пример

Ввод	Вывод
7 -4 9 8 4	24.005616

Примечания

Фигура представляет собой регион внутри Охраняющего Круга и внутри большего квадрата, сформированного диагональю d1, но за пределами меньшего квадрата, сформированного диагональю d2. Квадраты выровнены таким образом, что их диагонали параллельны осям x и y, а их центры точно совпадают с центром круга.



Решение

```
import math

def estimate_area(cx, cy, r, d1, d2, grid_size):
```

```
    x_min, x_max = cx - r - d1, cx + r + d1
    y_min, y_max = cy - r - d1, cy + r + d1

    delta = (x_max - x_min) / grid_size

    points_in_diamonds = 0
    total_points = 0

    for i in range(grid_size + 1):
        for j in range(grid_size + 1):
            x = x_min + i * delta
            y = y_min + j * delta

            # Check circle condition
            if (x - cx)**2 + (y - cy)**2 <= r**2:
                total_points += 1

                if abs(x - cx) + abs(y - cy) <= d1/2 and abs(x - cx) + abs(y - cy) > d2/2:
                    points_in_diamonds += 1

    total_area = math.pi * r**2
    area_per_point = total_area / total_points
    area_of_phi = points_in_diamonds * area_per_point

    return area_of_phi

cx, cy, r, d1, d2 = map(int, input().split())
area = estimate_area(cx, cy, r, d1, d2, grid_size=1000)
print(f"{area:.6f}")
```