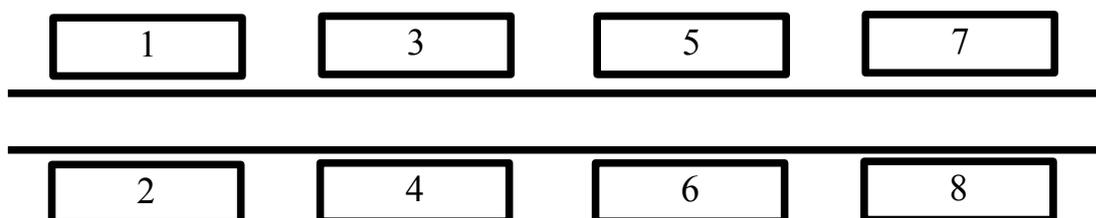


Задача 1. Улица

Автор задачи — Денис Кириенко

По одну сторону улицы находятся дома с нечётными номерами (1, 3, 5, ...), по другую сторону — с чётными (2, 4, 6, ...). Дом № 1 находится напротив дома № 2, дом № 3 — напротив дома № 4 и т. д. До соседнего дома нужно идти вдоль по улице одну минуту, независимо, с какой стороны улицы он находится (то есть от дома № 1 нужно идти одну минуту как до дома № 3, так и до дома № 4). До дома, стоящего напротив, идти не нужно.



Человек вышел на улицу из дома номер A и должен дойти до дома номер B . Определите, сколько минут ему нужно идти вдоль по улице.

Программа получает на вход два различных целых положительных числа A и B , не превосходящие 2×10^9 , — номера домов. Программа должна вывести одно число — искомое количество минут.

Пример входных и выходных данных

Ввод	Вывод
1 8	3

Система оценивания

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в 60 баллов.

Решение

Введём систему координат, направленную вдоль улицы. Пусть у дома номер 1 и 2 координата равна 1, у домов номер 3 и 4 координата равна 2 и т. д. Если номер дома равен k , то его координата будет равна $(k + 1) // 2$. Под «//» здесь подразумевается целочисленное деление (синтаксис соответствует языку Python), в языке Pascal это операция div.

Тогда если координата одного дома равна s , а другого дома — t , то расстояние между этими домами будет равно $\text{abs}(s - t)$.

Пример решения на языке Python.

```
a = int(input())
b = int(input())
print(abs((a + 1) // 2 - (b + 1) // 2))
```

Задача 2. Надёжное крепление

Автор задачи — Денис Кириенко

Уличный рекламный щит прикреплен к опоре при помощи трёх креплений. Первое крепление может выдерживать ветер, скорость которого не превосходит A м/с, второе крепление – B м/с, третье – C м/с. Сам щит будет надёжно закреплён, если как минимум два крепления из трёх выдерживают ветер данной скорости. Определите максимальную скорость ветра, которую выдержит данный щит.

Программа получает на вход три целых положительных числа A , B , C , не превосходящие 2×10^9 , – допустимые скорости ветра, которые выдерживают три крепления щита. Программа должна вывести одно число – максимальную скорость ветра, которую выдержит щит.

Пример входных и выходных данных

Ввод	Вывод
28 15 10	15

Система оценивания

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в 60 баллов.

Решение

Ответом является «медиана» – среднее из трёх чисел, если их упорядочить. Например, если для чисел A , B , C выполнены неравенства $A \leq B \leq C$, то ответом будет значение B . Реализовать алгоритм нахождения медианы можно разными способами. Пример решения, разбирающего разные способы упорядочить числа A , B , C .

```
a = int(input())
b = int(input())
c = int(input())
if a <= b <= c or c <= b <= a:
    print(b)
elif b <= a <= c or c <= a <= b:
    print(a)
else:
    print(c)
```

В следующем примере используется функция `min` для поиска наименьшего из трёх чисел, функция `max` для поиска наибольшего из трёх чисел, а медианой является оставшееся из трёх чисел, которое можно найти, вычтя из суммы данных чисел значение наибольшего и наименьшего числа.

```
a = int(input())
b = int(input())
c = int(input())
print(a + b + c - min(a, b, c) - max(a, b, c))
```

Можно воспользоваться стандартной функцией сортировки и вывести значение среднего элемента упорядоченного списка.

```
a = int(input())
b = int(input())
c = int(input())
print(sorted([a, b, c])[1])
```

Задача 3. Парад

Автор задачи — Роман Гусарев

В параде принимают участие M военных. Командование парада решило, что наиболее эффективное построение военных – в форме квадрата, то есть число участников построения должно быть точным квадратом. Но поскольку число M может не быть точным квадратом, разрешается разбить военных на несколько полков, каждый из которых строится в форме квадрата. Для красоты все полки должны быть одинакового размера, также командование парада хочет, чтобы размер каждого полка был как можно больше. Определите максимально возможный размер полка.

Программа получает на вход одно целое положительное число M , не превосходящее 2×10^9 , – количество участников парада. Программа должна вывести одно число – максимально возможный размер полка.

Пример входных и выходных данных

Ввод	Вывод
180	36

Система оценивания

Решение, правильно работающее только для случаев, когда M не превосходит 10000, будет оцениваться в 60 баллов.

Решение

В этой задаче нужно просто перебрать все возможные ответы. Нам нужно найти такое максимальное число K , которое было бы делителем числа M и полным квадратом: $K = d^2$. Между тем решение, которое будет перебирать все числа от 1 до M и проверять, является ли оно подходящим ответом, будет работать слишком долго. Такое решение будет набирать 60 баллов.

Для того, чтобы написать решение на 100 баллов, необходимо сразу же перебирать квадраты чисел (то есть значения d), до тех пор, пока рассматриваемый квадрат не станет больше числа M .

Пример решения на языке Python.

```
n = int(input())
ans = 1
d = 2
while d * d <= n:
    if n % (d * d) == 0:
        ans = d * d
    d += 1
print(ans)
```

Задача 4. Ряд чисел

Автор задачи — Роман Гусарев

Легенда гласит, что Карл Фридрих Гаусс, учась в школе, смог быстро посчитать

сумму целых чисел от 1 до 100, заметив, что $1 + 100 = 2 + 99 = \dots = 50 + 51$. Теперь решите задачу посложнее: можно ли перед каждым из чисел от 1 до N расставить знаки «+» или «-» так, чтобы сумма получившихся чисел была равна 0? Например, для $N = 3$ сумма $-1 -2 +3$ будет равна 0, а для $N = 2$ этого сделать нельзя.

Программа получает на вход целое неотрицательное число N , не превосходящее 10^5 . Программа должна вывести последовательность из N символов «+» или «-», соответствующих знакам, которые нужно расставить перед числами от 1 до N так, чтобы сумма получившихся чисел была равна 0. Если задача имеет несколько решений, нужно вывести один (любой) ответ. Если задача не имеет решения для данного N , нужно вывести одно слово «IMPOSSIBLE».

Примеры входных и выходных данных

Ввод	Вывод	Примечание
3	--+	Правильным ответом будет также «+-»
2	IMPOSSIBLE	

Система оценивания

Решение, правильно работающее только для случаев, когда N не превосходит 20, будет оцениваться в 40 баллов.

Решение

К идее решения задачи можно прийти, если попробовать построить ответы для маленьких значений N . При $N = 1$ и $N = 2$ задача не имеет решения, при $N = 3$ и при $N = 4$ есть решения («+-» и «+--»). Далее заметим, что для любых 4 подряд идущих чисел (не только для чисел 1, 2, 3, 4) можно расставить знаки требуемым образом: «+--». Поэтому если N делится на 4, то ответом будет строка «+--», повторённая $N / 4$ раза. Если N даёт остаток 3 при делении на 4, то ответ можно получить, записав сначала строку «+-», а затем повторив строку «+--» нужное число раз. Если N даёт остаток 1 или 2 при делении на 4, то задача не имеет решения. Для доказательства можно заметить, что в этом случае сумма чисел от 1 до N будет нечётной, а замена знака одного слагаемого не меняет чётность суммы, поэтому при любой расстановке знаков сумма будет нечётной.

В решении необходимо разобрать случаи различного остатка от деления N на 4 и для каждого случая вывести свой вариант ответа.

Пример решения на языке Python.

```
n = int(input())
if n % 4 == 1 or n % 4 == 2:
    print("IMPOSSIBLE")
elif n % 4 == 3:
    print("+-" + "+--" * (n // 4))
else:
    print("+--" * (n // 4))
```

Задача 5. Клад

Автор задачи — Денис Кириенко

Путь к кладу задан в виде указаний, какое количество шагов нужно пройти в одном из четырёх направлений: север (N), юг (S), запад (W), восток (E). Весь маршрут записан в виде строки, содержащей последовательность из чисел и следующих за числами букв, указывающих направление перемещения. Например, строка «7N5E2S3E» означает «пройти 7 шагов на север, 5 шагов на восток, 2 шага на юг, 3 шага на восток». В маршруте может быть много команд перемещения, поэтому каждый такой маршрут можно сократить.

Например, ранее приведённый маршрут можно сократить до «5N8E». По данному маршруту до клада сократите его до строки минимальной длины.

Программа получает на вход строку, состоящую из целых неотрицательных чисел, не превосходящих 10^7 каждое, и одной буквы («N», «S», «W», «E»), следующей за каждым числом. Других символов (в том числе пробелов), кроме цифр и букв направлений, в строке нет. Длина строки не превосходит 250 символов. Гарантируется, что начальная и конечная точки маршрута различаются.

Программа должна вывести маршрут, ведущий в ту же точку, записанный в таком же виде, как во входных данных, используя минимальное число символов. Если ответов несколько, программа должна вывести один (любой) из них.

Примеры входных и выходных данных

Ввод	Вывод	Примечание
7N5E2S3E	5N8E	Правильным ответом будет также «8E5N»
10N30W20N	30N30W	Правильным ответом будет также «30W30N»

Система оценивания

Решение, правильно работающее только для случаев, когда все числа во входной строке являются однозначными, будет оцениваться в 60 баллов.

Решение будет оцениваться в том случае, если оно проходит первый тест из условия (прохождение второго теста из условия обязательным не является).

Решение

Это — техническая задача, главную сложность в решении представляет разбор входной строки и выделение в ней чисел и команд перемещения.

Будем запоминать текущее положение в переменных `move_n` (число сделанных шагов на север) и `move_e` (число сделанных шагов на восток). При совершении перемещений на юг и на запад будет уменьшать эти переменные, их значение может стать отрицательным. При считывании одной из букв направления, мы будем менять соответствующую переменную на количество шагов, предшествовавших направлению.

Для определения количества шагов, предшествующих направлению, заведём переменную `num`, в которой будем хранить значение считанного числа. Если следующий символ строки будет цифрой, то к переменной `num` дописывается эта цифра в конец, для этого значение `num` нужно умножить на 10 и добавить значение новой считанной цифры. При считывании буквы значение `num` сбрасывается в 0.

После разбора всей строки нужно вывести текущее перемещение. Если значение `move_n` будет положительным, то нужно вывести значение `move_n` и букву «N». Если значение `move_n` будет отрицательным, то выведем значение `-move_n` и букву «S». При нулевом значении `move_n` не будет выведена ни буква «N», ни буква «S». Аналогично выведем часть ответа, соответствующую перемещению на восток или на запад.

Пример решения на языке Python.

```
s = input()
move_n = 0
move_e = 0
num = 0
for c in s:
    if c == 'N':
        move_n += num
        num = 0
    elif c == 'S':
```

```

        move_n -= num
        num = 0
    elif c == 'E':
        move_e += num
        num = 0
    elif c == 'W':
        move_e -= num
        num = 0
    else:
        num = num * 10 + int(c)
if move_n > 0:
    print(move_n, "N", sep='', end='')
if move_n < 0:
    print(-move_n, "S", sep='', end='')
if move_e > 0:
    print(move_e, "E", sep='')
if move_e < 0:
    print(-move_e, "W", sep='')

```

На языке C++ считывание данных можно реализовать ещё проще.

```

#include<iostream>
#include<map>
using namespace std;
int main()
{
    map<char, int> X {{'E', 1}, {'W', -1}};
    map<char, int> Y {{'N', 1}, {'S', -1}};
    int x = 0;
    int y = 0;
    int n;
    char dir;
    while (cin >> n >> dir)
    {
        x += n * X[dir];
        y += n * Y[dir];
    }
    if (x > 0) cout << x << "E";
    if (x < 0) cout << -x << "W";
    if (y > 0) cout << y << "N";
    if (y < 0) cout << -y << "S";
}

```

Здесь сразу считывается число и символ после него (`cin >> n >> dir`). Цикл продолжается, пока считывание успешно. В этом решении мы также использовали структуру данных «словарь» (`map`) для сопоставления каждой букве — числа (величине перемещения по оси OX и оси OY для каждого направления перемещения).