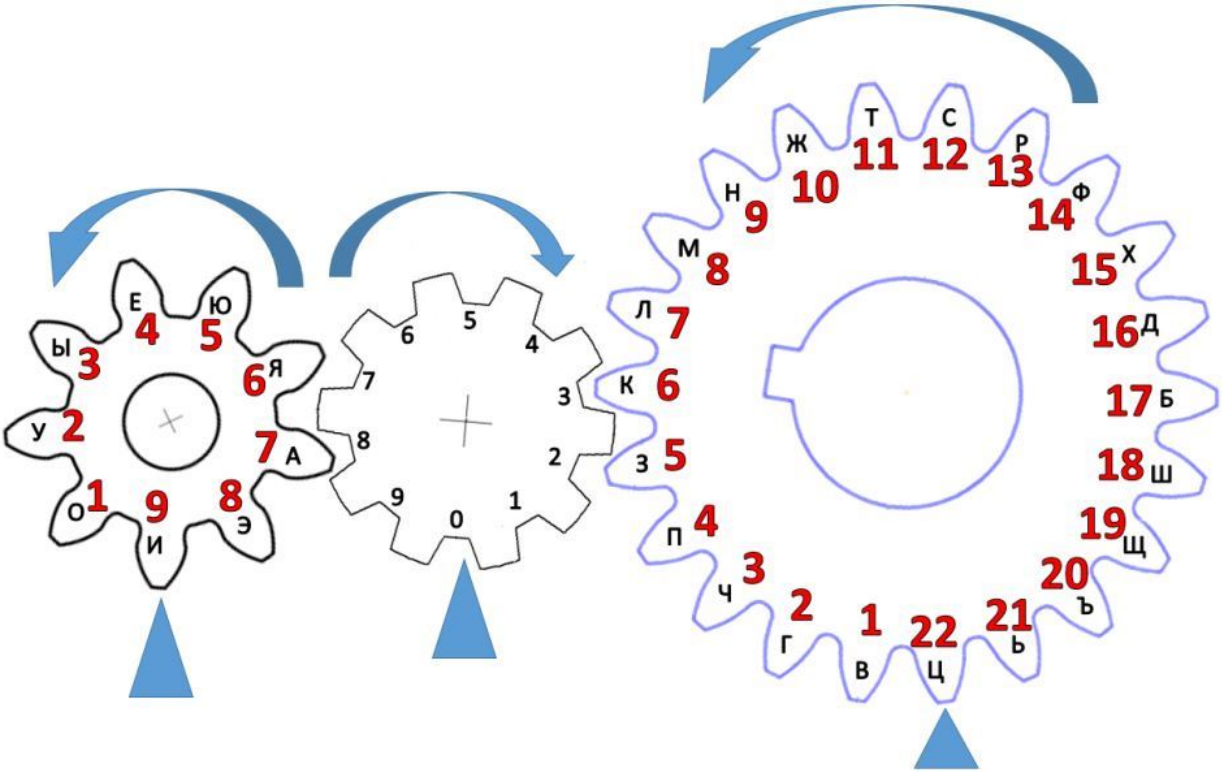
Задача 1. Полет на игру

Заметим, что для любых кресел A и B, расположенных через проход, величина $B - A$ будет одинаковой. Прибавив к разности $B - A$ количество кресел $K - 1$, получим величину, равную половине кресел в салоне самолета. Умножаем это число на 2 и получаем нужный результат. Формула выглядит так $(B - A + K - 1) * 2$.

Задача 2. Шифровка

Стоит заметить, что числа в шифровке идут не только по возрастанию, но и по убыванию. А так как колесо крутится только по часовой стрелке, это означает, что меньшее число могло выпасть только после перехода на следующий круг. Учитывая круги, предположим, что шифр 4365 представляет собой пары букв под номерами 4, 13, 16 и 25, которые отображаются на цифровом колесе. Буквенные колеса также можно пронумеровать.



Чтобы понять, какие номера букв выпали, необходимо четыре числа: 4, 13, 16 и 25 поделить с остатком на 9 для гласных букв, и на 22 — для согласных(по количеству букв в наборе). Получим следующие пары:

Номер	остаток от деления на 22	согласная буква	остаток от деления на 9	гласная буква	получившийся слог
4	4	П	4	Е	ПЕ
13	13	Р	4	Е	РЕ
16	16	Д	7	А	ДА
25	3	Ч	7	А	ЧА

Таким образом, правильный ответ слово `ПЕРЕДАЧА`.

Задача 3. Проблемы с двигателями

Для определения минимального потребления рассмотрим каждый двигатель, начиная со второго, так как первый не может быть заглушен по условию. Вместо энергопотребления каждого двигателя будем подставлять результат сложения значения его потребления с минимальным значением из двух предыдущих. Таким образом, моделируется ситуация, когда мы либо останавливаем предыдущий двигатель, либо не останавливаем его. Из первого примера $5-3-4-2-3$ получится следующий числовой ряд $5-8-9-10-12$. Рассмотрев остальные примеры, получаем наборы:

Номер примера	Потребление двигателей, единиц топлива	Сумма оставшихся
1	5 8 9 10 12	12
2	1 7 4 6 9 14	14
3	4 10 11 15 17 22 18 20 26	26
4	2 7 5 10 8 12 15 15 23 19 28 25 32	32

В следующей таблице приведены потребления двигателей после того, как заглушили некоторые из них. В последнем столбце таблицы расположен ответ на каждый пример задачи.

Номер примера	Потребление двигателей, единиц топлива	Сумма оставшихся
1	5 0 4 0 3	12
2	1 0 3 2 0 8	14
3	4 0 7 0 6 0 1 0 8	26
4	2 0 3 0 3 4 0 3 0 4 0 6 7	32

Задача 4. Вкусные конфеты

Заметим, что максимального количества конфет можно добиться, сначала применяя к выбранным карточкам первую операцию, а затем — вторую к оставшимся карточкам. Если суммы карточек, выбранных для первой и второй операций соответственно, равны X и Y, то в итоге тренер получит XY конфет. Так как сумма $X+Y$ зафиксирована, то произведение X и Y окажется максимальным, когда X и Y будут наиболее близки друг к другу. Остается найти оптимальное разбиение для трех тестов.

Номер набора	числа на карточках	Максимальное количество конфет
1	2 3 4 5 6	$(2+3+5)\cdot(4+6)=100$
2	2 3 4 5 6 7	$(2+3+4+5)\cdot(6+7)=182$
3	2 3 4 5 6 7 8	$(4+6+8)\cdot(2+3+5+7)=306$

Задача 5. Стеф и его команда

Рассмотрим время, оставшееся до конца матча, которое составляет $48\cdot 60-60\cdot m-s$ секунд. Разобьем его, начиная с текущего момента, на интервалы по 4 атаки, т.е. по $24\cdot 4$ секунды. В конце матча, возможно, интервал будет неполным. В любом полном интервале команда «Воины золотого штата» совершит две успешных атаки и получил суммарно 6 очков, а команда «Востон Келтикс» совершит две атаки, только одна из которых будет успешной и принесет спортсменам 2 очка. Количество таких полных интервалов равно неполному частному от деления оставшегося времени, выраженному в секундах, на $24\cdot 4=96$. Таким образом, к текущему счету следует прибавить 6 и 2 соответственно, умноженные на количество полных интервалов.

Осталось разобраться с неполным интервалом. Его продолжительность равна остатку от деления оставшегося времени на $24\cdot 4=96$. Если промежуток продлится хотя бы 24 секунды, то команда «Воины золотого штата» успеет завершить свою атаку и получит 3 очка, которые нужно добавить к результирующему счету. Если продолжительность неполного интервала хотя бы $2\cdot 24$ секунды, то команда «Востон Келтикс» также успеет совершить свою успешную атаку и получит плюс 2 очка к итоговому результату. Наконец, если этот интервал не короче, чем $3\cdot 24$ секунды, то команда «Воины золотого штата» успеет завершить и вторую свою атаку — добавляем еще 3 очка к ее счету.

Готовое решение на Python выглядит следующим образом:

```python
m = int(input())
s = int(input())
p = int(input())
q = int(input())
```

```
time = 48 * 60 - 60 * m - s
k = time // (24 * 4)
p += k * 6
q += k * 2
time %= 24 * 4
if time >= 24:
    p += 3
if time >= 24 * 2:
    q += 2
if time >= 24 * 3:
    p += 3
print(p, q)
```

Задача 6. Гонка за лидером

Чтобы понять, сможет ли Стеф догнать своего конкурента, необходимо сравнить текущую разницу между ними и количество очков, на которое спортсмен может уменьшить разрыв до конца сезона. Если разница больше, то вывести «No». В противном случае, необходимо еще выяснить количество игр, необходимых Стефу для этого. Для этого мы поделим с округлением вверх количество очков, необходимое для выполнения задачи T на количество очков S, набираемых Стефом за каждую игру.

Готовое решение на Python выглядит следующим образом:

```
K = int(input())
L = int(input())
N = int(input())
S = int(input())
S //= 2
T = L - K
if T > N * S:
    print('No')
else:
    print('Yes')
    print((t + s - 1) // s)
```

Задача 7. Последовательность

Заметим, что отличаться ровно на x могут только номера матчей, дающих одинаковые остатки по модулю x. Таким образом, достаточно для каждого остатка k $(0 \leqslant k < x)$ найти максимальное количество натуральных чисел, не превосходящих n, чтобы разность никаких двух из которых не равнялась x. Для этого достаточно взять числа через одно, т.е. k, $k+2x$, $k+4x$,... до тех пор, пока очередной номер не станет больше n. Очевидно, что данная подпоследовательность обеспечивает максимальное количество выбранных номеров, однако в некоторых случаях она не является единственно возможной — допустимо также взять подпоследовательность $k+x$, $k+3x$, $k+5x$,... если в ней такое же количество элементов, не превосходящих n. Если для всех k выбрать первый вариант подпоследовательности, то итоговый ответ будет выглядеть так: $1, 2, \ldots, x, 2x+1, 2x+2, \ldots, 3x, 4x+1, 4x+2, \ldots, 5x, \ldots$

Готовое решение на Python выглядит следующим образом:

```
n = int(input())
x = int(input())
for i in range(n):
    if i // x % 2 == 0:
        print(i + 1)
```