Задача A. Петя в Школе

В первом случае можно прийти в дни с номерами: $1, 3, 4, 5$. Таким образом будет пропущен только один день.

Во втором примере можно прийти в дни: $1, 2, 4, 5, 7, 8, 9, 10, 11$ и пропустить всего три дня.

В третьем примере можно пропустить дни с номерами: $7, 14, 21, 28, 35$ и посетить школу в остальные дни.

В четвертом примере можно пропустить дни с номерами: $9, 18, 27, 36, 45, 54$.

Ответ на задачу: 1 3 5 6.

Задача B. Вирусы

Сделаем некоторые наблюдения. Пусть изначальное количество елок равно n, а ночью были сломаны t елок.

Тогда изначальное количество гирлянд равно $\frac{n \cdot x}{2}$, а количество гирлянд в конце — $\frac{(n-t) \cdot y}{2}$.

Составим уравнение: $\frac{n \cdot x}{2} - z = \frac{(n-t) \cdot y}{2}$. Для удобства домножим обе части на два и раскроем скобки. Получим выражение: $n \cdot (x + y) + t \cdot y = 2z$.

При этом должно быть выполнено неравенство $n \geqslant t$, а также числа $n \cdot x$ и $(n - t) \cdot y$ должны быть четными.

Теперь заметим, что перебрать все подходящие решения достаточно просто.

Рассмотрим первый пример. Уравнение имеет вид: $2n + 5t = 80$. Для переменной n подходят значения $5, 10, 15, 20, \ldots$. Рассмотрев несколько вариантов легко понять, что максимальное значение t равно 8.

Аналогично рассмотрим оставшиеся два случая.

Ответ: 8 5 8.

Задача C. Клава и Оригами

Во-первых, Клавдии потребуется потратить одну минуту, чтобы сделать каждую из N поделок, поэтому сразу прибавим к ответу N. Осталось посчитать, сколько минут девочка потратит на отдых.

Для начала определим, сколько раз Клавдия будет отдыхать. Так как она отдыхает после каждых сделанных M поделок, то она будет отдыхать ровно $K = \left\lceil \frac{N}{M} \right\rceil - 1$ раз.

Теперь посчитаем суммарное время отдыха. Оно равно $1 + 2 + 3 + \ldots + K = \frac{K(K+1)}{2}$.

Таким образом, ответ равен $N + \frac{K(K+1)}{2}$.

Асимптотика решения: $\mathcal{O}(1)$.

Пример решения на языке C++:

```cpp
#include <cstdio>

using namespace std;

using ll = long long;

int main() {
    ll n, m;
    scanf("%lld%lld", &n, &m);

    ll k = (n + m - 1) / m - 1;

    printf("%lld\n", n + k * (k + 1) / 2);

    return 0;
}
```

Задача D. Маски

Рассмотрим всевозможные варианты закупки масок и для каждого посчитаем стоимость. После этого выберем способ закупки с минимальной суммарной стоимостью масок.

1. Каждый ученик будет использовать три одноразовых маски каждый день (это минимальное количество масок, которого хватит). Суммарная стоимость равна $3 \cdot n \cdot k \cdot c_1$.
2. У каждого ученика будет одна многоразовая маска на все дни, а также дополнительная одноразовая маска на каждый день. Суммарная стоимость: $n \cdot k \cdot c_1 + n \cdot c_2$.
3. У каждого ученика будут две многоразовые маски. Стоимость равна: $2 \cdot n \cdot c_2$.

Чтобы получить ответ, необходимо вычислить минимум из данных трех величин. Не забудьте воспользоваться 64-битным типом данных.

Асимптотика решения: $\mathcal{O}(1)$.

Пример решения на языке C++:

```
#include <cstdio>
#include <algorithm>

using namespace std;

using ll = long long;

int main() {
    ll n, k, c1, c2;
    scanf("%lld%lld%lld%lld", &n, &k, &c1, &c2);

    ll ans1 = 3 * n * k * c1;
    ll ans2 = n * k * c1 + n * c2;
    ll ans3 = 2 * n * c2;

    printf("%lld\n", min(min(ans1, ans2), ans3));

    return 0;
}
```

Задача E. Разделяй и Властвуй

В данной задаче необходимо понять, сколько разрезов будет сделано. Пусть у нас имеется квадрат $N \times N$. Пусть $f(N)$ — количество разрезов, которые необходимо сделать. Рассмотрим несколько случаев:

1. $N = 1$. В этом случае, очевидно, делать разрезы не нужно, то есть $f(N) = 0$.
2. $N = 2k$, то есть число N — четное. В этом случае мы сделаем два разреза, после чего нам нужно будет разрезать на кусочки четыре квадрата размера $\frac{N}{2} \times \frac{N}{2}$. То есть $f(N) = 2 + 4 \cdot f\left(\frac{N}{2}\right)$.
3. $N = 2k + 1$, то есть число N — нечетное. В этом случае будут сделаны два разреза, после которых останется таблица $(N-1) \times (N-1)$, которую нужо будет разрезать в дальнейшем. То есть $f(N) = 2 + f(N-1)$.

Вычислим ответ по полученной формуле. Следует не забыть воспользоваться 64-битным типом данных.

Асимптотика решения: $\mathcal{O}(\log N)$, так как каждый второй раз мы уменьшаем размер стороны таблицы в два раза.

Пример решения на языке C++:

```
#include <cstdio>

using namespace std;

using ll = long long;

ll f(ll n) {
    if (n == 1) {
        return 0;
    }
    if (n % 2 == 0) {
        return 4 * f(n / 2) + 2;
    } else {
        return f(n - 1) + 2;
    }
}

int main() {
    ll n;

    scanf("%lld", &n);
    printf("%lld\n", f(n));

    return 0;
}
```