1. Решите уравнения в целых числах, используя расширенный алгоритм Евклида. Внимание! Требуется найти все решения, а не только частное решение, которое находит алгоритм Евклида. Выведите самостоятельно формулу для общего решения или воспользуйтесь помощью литературы.

а) $238x + 385y = 133$

a	b	$238x + 385y$
1	0	238
-1	1	147
2	-1	91
-3	2	56
5	-3	35
-8	5	21
13	-8	14
-21	13	7
55	-34	0

$\frac{133}{7} = 19 \Rightarrow x = -399, y = 247 \Rightarrow (-399, 247)$ – **частное решение**.
Общее решение: $(55n - 399, -34n + 247)$

б) $143x + 121y = 52$.

a	b	$143x + 121y$
1	0	143
0	1	121
1	-1	22
-5	6	11
11	-13	0

НОД $(143, 121) = 11$, но 52 не делится на 11 без остатка, поэтому в целых числах уравнение **не имеет решений**.

2. Решите сравнение $68x + 85 \equiv 0 \pmod{561}$ с помощью расширенного алгоритма Евклида. (Требуется найти все решения в вычетах)

Solution

68x $\equiv -85(mod561)$
68x $\equiv 476(mod561)$
68x + 561y = 1

А такое уравнение мы умеем решать с помощью алгоритма Евклида!

a	b	$68x + 561y$
0	1	561
1	0	68
-8	1	17
33	-4	0

$-266x + 33y = 561$
$-266x \equiv 0(mod 33)$
Ответ: $x = 33n, n \in 0, 1, ...16$

3. Вычислите $7^{13} \mod 167$, используя алгоритм быстрого возведения в степень.
$7^{13} \equiv 7 \cdot (7^6)^2 \equiv 7 \cdot ((7^3)^2)^2 \equiv 7 \cdot ((7 \cdot 7^2)^2)^2 \equiv 7 \cdot (81)^2 \equiv -2(mod 167)$

4[ДПВ 1.8]. Доказать корректность рекурсивного алгоритма умножения Divide (раздел 1.1., рис. 1.2.) и получить верхнюю оценку на время работы.

Корректность следует из того, что $y \geqslant 1$, а также из того, что функция деления (без нуля в знаменателе) очень похожа на функцию умножения, корректность которой была доказана ранее.

Верхняя оценка для вычисления времени работы – $O(n^2)$. Считается аналогично нахождения сложности работы алгоритма умножения, который был рассмотрен ранее.

5. Функции $T_1(n)$ и $T_2(n)$ заданы рекуррентными формулами, известно что $T_i(1) = T_i(2) = T_i(3) = 1, i = 1, 2$.

1. Найдите асимтотику роста функции $T_1(n) = T_1(n-1) + cn$ (при $n > 3$);
$\Theta(n)$

2. Докажите, что для функции $T_2(n) = T_2(n-1) + 4T_2(n-3)$ (при $n > 3$) справедлива оценка $\log T_2(n) = \Theta(n)$.

Это доказывается с помощью построения дерева рекурсивных вызовов: поскольку каждый вызов активирует вместе с собой ещё два вызова, то высота дерева будет равна двоичному логарифму от n.

3*. Найдите (точную) асимтотику роста функции $T_2(n)$.

6 [Шень 1.1.17]. Добавим в алгоритм Евклида дополнительные переменные `u`, `v`, `z`:

```
 m := a; n := b; u := b; v := a;
{инвариант: НОД (a,b) = НОД (m,n); m,n >= 0 }
while not ((m=0) or (n=0)) do begin
| if m >= n then begin
| | m := m - n; v := v + u;
| end else begin
| | n := n - m; u := u + v;
| end;
end;
if m = 0 then begin
| z:= v;
end else begin {n=0}
| z:= u;
end;
```

Докажите, что после исполнения алгоритма значение `z` равно удвоенному наименьшему общему кратному чисел `a`, `b`: `z` $= 2 \cdot$ `HOK(a,b)`.

Для начала стоит отметить, что $m \cdot u + n \cdot v$ не меняется в ходе алгоритма и вначале равна ab.

Далее замечаем, что НОД(a, b) ·

7*. Предложите $O(\sqrt{m}\log m)$ алгоритм нахождения длины периода десятичной дроби $\frac{n}{m}$. Докажите его корректность и оцените асимптотику.

8*. Доказать, что `inv(i, p): return i > 1 ? -(p/i)*inv(p%i, p) % p : 1` возвращает обратный остаток, доказать, что работает за логарифм и развернуть рекурсию.

9*. $f(1) = g(1) = 1$ $f(n) = a \cdot g(n-1) + b \cdot f(n-1)$ $g(n) = c \cdot g(n-1) + d \cdot f(n-1)$ где a, b, c, d положительные константы. Предложите алгоритм вычисляющий $f(n)$ со сложностью $O(\log n)$ арифметических операций.