В скобках к задачам указаны баллы. Если баллы не указаны, то задачу можно не сдавать (как и задачи со звёздочкой).

1 (2). Решите[1] уравнения в целых числах, используя расширенный алгоритм Евклида:

а) $238x + 385y = 133$; **б)** $143x + 121y = 52$.

2 (2). Вычислите $7^{13} \mod 167$, используя алгоритм быстрого возведения в степень.

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

4 (3). Опишем преобразование $\mathtt{SX}(m)$ двоичной записи числа m. Каждая единица заменяется на SX, а ноль на S, после чего вычёркивается первая слева S. Так $(5) = 101 \to XSSX$. Алгоритм вычисления функции от положительных целых чисел (x, m) задан псевдокодом:

```
Function F(x, m) :
    a = SX(m);
    y = 1;
    for i = 1 to a.size do
        if a[i] == X then
            y = y × x
        else
            y = y × y
        end
    end
    return y
end
```

1. Вычислите $\mathtt{F}(3, 5)$.

2. Какую математическую функцию реализует данный алгоритм?

3. Докажите корректность данного алгоритма (нужно доказать, что алгоритм действительно реализует отгаданную функцию).

4. Оцените время работы алгоритма, считая, что арифметические операции стоят $O(1)$.

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

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

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

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

6 (4). В низкоуровневом языке программирования используются регистры, в которых хранятся двоичные последовательности одинаковой, но произвольной длины. С регистрами разрешены следующие операции: 1: изменить значение первого бита, 2: изменить значение бита, стоящего после первой единицы. Постройте алгоритм, который получив на вход содержимое двух регистров пишет код, реализующий копирование содержимого первого регистра во второй и оцените сложность этого алгоритма.

[1] Вы должны найти все решения!

Вам доступно только два регистра P и Q. Вход: содержимое регистров P и Q (числа в двоичной записи). Выход: код с командами вида $R1$ и $R2$, где R — регистр (P или Q), а 1 и 2 номер соответствующей операции. В результате исполнения кода в регистре Q должно оказаться значение регистра P (при условии её запуска на содержимом регистров со входа). При построении алгоритма, который строит код, Вы работаете в стандартной модели вычислений (арифметические операции стоят $O(1)$).

7 [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` $= 2 \cdot$ `НОК(a,b)`.

8*. Предложите полиномиальный алгоритм нахождения периода десятичной дроби $\frac{n}{m}$. Докажите его корректность и оцените асимптотику.

9*. 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.

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