1 Рассмотрим каждый из пунктов:

а) Решаем след. ур-ние:

$$238 * x + 385 * y = 133$$

Составим таблицу:

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

Как мы видим, правая часть уравнения ($C = 133$) делится нацело на НОД(238;385) $= 7 \Rightarrow$ можем записать частное решение ур-ния:

$$\begin{cases} x_0 = (133 : 7) * (-21) = -399, \\ y_0 = (133 : 7) * 13 = 247 \end{cases}$$

Теперь найдем общее решение. Разделим обе части исходного ур-ния на НОД $=$ 7. Для нахождения общего решения воспользуемся фактом:

$$A * x + B * y = C \Rightarrow A * (x - k * B) + B * (y + k * A) = C, \forall k \in \mathbb{Z}$$

Таким образом, общее решение буде иметь вид:

$$\begin{cases} x = x_0 + \frac{B}{(A;B)} * k, \\ y = y_0 - \frac{A}{(A;B)} * k \end{cases}$$

$$\begin{cases} x = -399 + 55 * k, \\ y = 247 - 34 * k \end{cases}$$

б) Решаем след. ур-ние:

$$143 * x + 121 * y = 52$$

Составим таблицу:

x	y	$143 * x + 121 * y$
1	0	143
0	1	121
1	-1	22
-5	6	11

Как мы видим, правая часть уравнения ($C = 52$) не делится нацело на НОД(143;121) $= 11 \Rightarrow$ это ур-ние не имеет целых решений.

2 Мы решаем сравнение

$$68 * x + 85 \equiv 0 (mod\, 561)$$

Это сравнение эквивалентно след. уравнению:

$$68 * x + 561 * y = 85$$

Составим таблицу:

x	y	$68 * x + 561 * y$
0	1	561
1	0	68
-8	1	17

По аналогии с прошлым заданием,

$$\begin{cases} x = -7 + 33 * k, \\ y = 1 - 4 * k \end{cases}$$

3 Нам необходимо вычислить $7^{13}\ mod\ 167$

$7^1 = 7 \equiv 7\ (mod\ 167)$
$7^3 = (7^1)^2 * 7 = 49 * 7 = 343 \equiv 9\ (mod\ 167)$
$7^6 = (7^3)^2 \equiv 9^2\ (mod\ 167) \equiv 81\ (mod\ 167)$
$7^{13} = (7^6)^2 * 7 \equiv 81^2 * 7\ (mod\ 167) \equiv 2\ (mod\ 167)$

Таким образом, используя алгоритм быстрого возведения в степень, мы получили, что $7^{13} \equiv 2\ (mod\ 167)$

4 Пройдемся по пунктам задачи отдельно:

1. а) Для начала покажем, что рекурсия остановится. На кадом шаге рекурсии вызывается эта же ф-я от аргумента $\lfloor \frac{x}{2} \rfloor$. Эта операция каждый раз отбрасывает младший бит (так работает округление вниз) $\Rightarrow$ рано или поздно будет вызвана ф-я от аргумента $x = 1$, а значит при следующем вызове будет $x = 0$, и рекурсия остановится.

б) Теперь покажем, что наш алгоритм приведет нас к верному результату. Воспользуемся ММИ. Пусть ф-я от аргументов $(\lfloor \frac{x}{2} \rfloor; y)$ возвращает верный результат. Тогда, рассмотрим 2 ситуации: x в данном локальном пространстве четно или нечетно. Если оно четно, то достаточно возвращаемые $(q; r)$ просто домножить на 2 и посмотреть на число $2*r$: взять его по модулю y (конечно же не забывая увеличить счетчик q в случае, если $2 * r \geqslant y$). Если оно нечетно, то в целом все действия будут проделаны так же, за исключением того что нужно будет к числу $2*r$ еще прибавить единицу, потому что при делении с округлением вниз эта единица "пропадает". Таким образом, наша функция вернет верный результат, а именно в качестве $q = 2 * q' + 2 * r'\ mod\ y$, а в качестве $r = 2 * r'\ mod\ y$. И таким способом получим разложение $x = q * y + r$, что и будет верным ответом на задачу.

2. Теперь оценим время работы $T(n)$ алгоритма. Имеет смысл рассматривать бинарную модель, так как мы работаем с числом в двоичном представлении (на каждом шаге отбрасываем младший бит). Время работы данного алгоритма имеет квадратичную ассимптотику, и вот, почему. Глубина рекусии в данном случае это $n = \log x$, и она равна длине входа. Так как мы рассматриваем бинарную модель, каждая операция при каждом вызове имеет время работы $O(n)$. Таким образом, $T(n) = O(n^2)$ (n - кол-во шагов и на каждом шаге сложность каждой операции $O(n)$).

5. Рассмотрим каждый пункт задачи отдельно:

1. Мы рассматриваем ф-ю $T_1(n) = T_1(n-1) + cn(n > 3)$. Выразим $N_1(n)$ через младшие ф-ии.

$$T_1(n) = T_1(n-1) + cn = T_1(n-2) + c(n-1) + cn = T_1(n-3) + c(n-2) + c(n-1) + cn = ... =$$

$$= T_1(n-k) + kcn - \frac{k*(k-1)}{2}c$$

Мы будем проделывать такое до тех пор, пока $k < n-3$. Как только $k = n-3$ получится, что $T_1(n) = O(n^2)$. Это мы, по сути, получили оценку на саму ф-ю $T_1(n)$. Если честно, я не до конца понял, что именно нужно посчитать, поэтому еще сделаю, что подумал немного более точно подойдет под формулировку задачи. А именно, рассмотрел асимптотику роста ф-ии, то есть $\frac{T_1(n)-T_1(n-1)}{n-(n-1)} = cn = O(n)$. Но это кажется слишком очевидеым и простым, поэтому я провел рассуждения выше.

2. Для доказательсва данного утверждения воспользуемся ММИ. Пусть верна оценка $\forall k < n$. Тогда, получаем:

$$T_2(n) = T_2(n-1) + 4 * T_2(n-3) \leqslant c_1 * 2^{n-1} + c_1 * 2^2 * 2^{n-3} = c_1 * 2^{n-1} = O(2^n)$$

$$T_2(n) = T_2(n-1) + 4 * T_2(n-3) \geqslant c_2 * 2^{n-1} + c_2 * 2^2 * 2^{n-3} = c_2 * 2^{n-1} = \Omega(2^n)$$

Таким образом, $\log T_2(n) = \Theta(n)$.