0.3 Как факторизация произвольного полинома сводится к задаче факторизации полиномов без кратных сомножителей

Эта самая легкая часть алгоритма. Решение известно из курса матанализа:
если p(x) - полином с переменной x, то полином
$p_1(x) = (p(x), p(x)')$ содержит все сомножители p(x), у которых кратность больше 1,
$p_2(x) = (p_1(x), p_1(x)')$ содержит все сомножители p(x), у которых кратность больше 2
и так далее.

Легко написать такую программу
toSquareFreePols(p),
которая получает на входе полином
input : $\mathbf{p(x)}$, а возвращает вектор
output : $[\mathbf{f_1(x), f_2(x), .., f_k(x), n_1, n_2, ..n_k}]$, такой что
$\mathbf{p(x) = f_1(x)^{n_1} f_2(x)^{n_2} * .. * f_k(x)^{n_k}}$.
Здесь n_j – это натуральные числа в порядке возрастания, $f_j(x)$- полиномы.
В качестве первого примера возьмите полином $x^6 + 4x^5 + 4x^4 = x^4(x+2)^2$.

$SquareFree$ – это стандартное название для полиномов, которые не содержат кратных множителей.

Действительно, ведь не будет ни одного сомножителя, который появляется вместе со своим квадратом.

РЕКОМЕНДАЦИЯ : **toSquareFreePols(p)**
Пусть $n = \mathbf{degree}(p(x))$ – степень полинома p. Заготовьте два вектора: $pols = \mathbf{O}_n; numbs = \mathbf{O}_n$;
для сохранения полиномов после дифференцирования и вычисления GCD.

А перед тем как завершить процедуру создайте вектор нужного размера $2k$, перепешите в него результат и верните его.

Пришлите свою рабочую программу и 2 примера – этот и еще один свой!

Примеры располагаются в области комментария в начале программы – примерно как javadoc перед процедурой.

END
out :

END
