ВТОРО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 7 МАЙ, 2017 Г. , ГРУПА А

Задача АК5. УМНОЖЕНИЕ НА ПОЛИНОМИ

Автор: Йордан Чапъров

Вашата задача е да напишете програма **`polymul`**, която генерира алгоритми за умножение на два полинома с фиксирани степени N и M.

Един такъв алгоритъм $A(N, M)$ получава на вход два полинома $P(x)$, $Q(x)$, представени чрез коефициентите си:

$P(x) = a_0 + a_1 * x + a_2 * x^2 + \dots + a_N * x^N$.

$Q(x) = b_0 + b_1 * x + b_2 * x^2 + \dots + b_M * x^M$.

На изход, $A(N, M)$ трябва да изведе $N+M+1$ числа $c_0, c_1, \dots, c_{N+M}$ - коефициентите на полином $R(x) = c_0 + c_1 * x + \dots + c_{N+M} * x^{N+M}$, такъв че

$R(x) = P(x) * Q(x)$ за всяко x.

Алгоритъмът представлява списък от аритметични операции за събиране (+), изваждане (-) и умножение (*). Всяка аритметична операция се използва върху две стойности, които може да са или входни коефициенти (a_K, b_K), или стойности, изчислени на предходна стъпка (отбелязваме ги като i_K), или константи K.

Примерен алгоритъм A(1, 1):

Стъпка	**Пояснение**
	Начало. Имаме на разположение стойности a_0, a_1, b_0, b_1.
`= a0 * b0`	Изчисляваме междинна стойност $i_1 = a_0 * b_0$
`= a1 * b1`	Изчисляваме междинна стойност $i_2 = a_1 * b_1$
`= a0 * b1`	Изчисляваме междинна стойност $i_3 = a_0 * b_1$
`= a1 * b0`	Изчисляваме междинна стойност $i_4 = a_1 * b_0$
`= i3 + i4`	Изчисляваме междинна стойност $i_5 = i_3 + i_4$ Използваме междинни стойности за аритметична операция (+) . $i_5 = (a_0 * b_1) + (a_1 * b_0)$
`= a0 * 34`	Изчисляваме междинна стойност $i_6 = a_0 * 34$ Умножаваме входен коефициент по константа. Това умножение въобще не влияе на резултата от алгоритъма – то е сложено само като пример за **тривиално** умножение (вижте по-долу).
`o i1 i5 i2`	Прекратяваме алгоритъма с коефициенти: $c_0 = i_1 = a_0 * b_0$, $c_1 = i_5 = (a_0 * b_1) + (a_1 * b_0)$, $c_2 = i_2 = a_1 * b_1$

Алгоритъмът за умножение на полиноми, който вашата програма генерира трябва да е не само верен, но и да използва възможно най-малко **нетривиални** умножения. Едно умножение наричаме **тривиално**, ако поне единият от операндите му е константа. Алгоритъмът A(1, 1) по-горе използва 4 нетривиални умножения ($a_0 * b_0$, $a_1 * b_1$, $a_0 * b_1$, $a_1 * b_0$) и едно тривиално умножение ($a_0 * 34$).

ВТОРО КОНТРОЛНО СЪСТЕЗАНИЕ
НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ВЕЛИКО ТЪРНОВО, 7 МАЙ, 2017 Г. , ГРУПА А

Да разгледаме по-добър алгоритъм за умножение на два полинома от степен 1:

Стъпка	**Пояснение**
	Начало. Имаме на разположение стойности a_0, a_1, b_0, b_1.
`= a1 + a0`	$i_1 = a_1 + a_0$
`= b1 + b0`	$i_2 = b_1 + b_0$
`= i1 * i2`	$i_3 = i_1 * i_2 = (a_1 + a_0) * (b_1 + b_0)$
`= a0 * b0`	$i_4 = a_0 * b_0$
`= a1 * b1`	$i_5 = a_1 * b_1$
`= i3 - i4`	$i_6 = i_3 - i_4 = a_1 * (b_1 + b_0) + a_0 * b_1$
`= i6 - i5`	$i_7 = i_6 - i_5 = a_1 * b_0 + a_0 * b_1$
`o i4 i7 i5`	

Този алгоритъм използва само 3 нетривиални умножения – по-добре от първона-чалния алгоритъм.

За да се избегнат проблеми с точността на дробните числа, ще работим по модул 1 000 000 007. Това означава, че всяка константа в алгоритъма трябва да е цяло число между 0 и 1 000 000 006. Всяка аритметична операция в алгоритъма, който генерирате, ще бъде сметната по модул 1 000 000 007.

Вход

На единствения ред на стандартния вход са зададени две числа N, M - степените на полиномите, които вашият алгоритъм трябва да умножи.

Изход

На стандартния изход изведете алгоритъм за умножение на два полинома $P(x)$ с коефициенти a_k и степен N, и $Q(x)$ с коефициенти b_k и степен M.

Всеки ред (с пореден номер L; номерата започват от 1) от алгоритъма, освен последния, трябва да изглежда като:

`= S (operation) T`, където *(operation)* е една от +, -, *, и S, T са:

- входни коефициенти ($\boldsymbol{a_K}$, $0 \le K \le N$ или $\boldsymbol{b_K}$, $0 \le K \le M$),
- междинни стойности ($\boldsymbol{i_K}$, $K < L$), или
- константи ($\boldsymbol{K}$: $0 \le K \le$ 1 000 000 006)

Междинните стойности i_k получават индекса си по реда, по който се изчисляват, започвайки от 1 .

Последният ред от алгоритъма трябва да изглежда като:

`o` J_0 J_1 ... J_{N+M} , където първият символ е малката латинска буква „o“, а $J_0, J_1, \ldots, J_{N+M}$ са стойности, които след изчисляване на всички стъпки от алгоритъма, ще бъдат в този ред коефициентите $c_0, c_1, \ldots, c_{N+M}$ на полином $R(x)$, такъв че: $R(x) = P(x) * Q(x)$. Всички елементи на изходните редове (оператори, операнди и символът „o“ на последния ред) са отделени един от друг с по един интервал.

Ограничения

$1 \leq N, M \leq 100$

Максимален брой аритметични операции в алгоритъма: 200 000

Коефициентите на полиномите *P(x), Q(x)* са цели числа между 0 и 1 000 000 006.

Оценяване

За всеки тест ще получите (0.228907 * ln(80.1263 * best / cost)) * (брой точки, предвидени за теста), където *best* е теоретичният минимум за броя нетривиални умножения, които алгоритъм за умножение на полиноми може да използва, а *cost* е броят нетривиални умножения, които вашият алгоритъм използва. Тук **ln** е натурален логаритъм.

Пример

Вход	**Изход**

`1 1`	`= a1 + a0`
	`= b1 + b0`
	`= i1 * i2`
	`= a0 * b0`
	`= a1 * b1`
	`= i3 - i4`
	`= i6 - i5`
	`o i4 i7 i5`