

Problem A. A Mimada

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

Saori é uma menina rica e mimada pelo seu avô, Sr. Kido. Como o aniversário de Saori está próximo, ela entregou uma lista de presentes para o seu mordomo Tatsumi. Quando Sr. Kido soube da lista mandou que Tatsumi contasse, para certas quantias de dinheiro, quantos subconjuntos de presentes resultam em um custo total igual a cada uma das quantias de dinheiro. Tatsumi deve considerar que cada presente pode ser comprado no máximo uma vez e que no máximo dez presentes diferentes apresentam um mesmo custo.

Quando Tatsumi terminou sua tarefa, a mimada Saori mostrou-se insatisfeita com sua lista original e decidiu retirar alguns presentes que não lhe agradavam mais. Isso preocupou Tatsumi, pois ele havia perdido a cópia original da lista de presentes e não queria que Sr. Kido descobrisse. Cabe a você ajudar Tatsumi a atualizar a contagem de subconjuntos de presentes, agora removendo os presentes descartados por Saori, para cada quantia de dinheiro que Sr. Kido havia solicitado a contagem.

Input

A primeira linha irá conter o inteiro N ($2 \leq N \leq 10^5$) que representa o número de quantias de dinheiro que possuem pelo menos um subconjunto não-nulo cujo custo total seja igual à quantia. As próximas N linhas possuem dois números inteiros cada a e b ($1 \leq a \leq 10^5, 1 \leq b \leq 10^{18}$) que representam, respectivamente, uma quantia de dinheiro e o número de subconjuntos de presentes cujo custo total é igual a essa quantia (todas as N quantias de dinheiro são distintas). A próxima linha contém um número inteiro M ($M \geq 1$) que representa o número de presentes que Saori deseja retirar da lista. As próximas M linhas contêm, cada uma, o nome do presente, representado por uma string de até dez letras minúsculas do alfabeto, e seu custo, representado por um número inteiro V ($1 \leq V \leq 10^5$). É garantido que sobrará pelo menos um presente na lista final.

Output

Para cada quantia de dinheiro que resulte em pelo menos um subconjunto não-vazio de brinquedos, imprima a quantia de dinheiro e o número de subconjuntos não-vazios de brinquedos, separados por um espaço. Estes pares devem ser impressos um por linha e devem estar em ordem crescente da quantia de dinheiro.

Example

standard input	standard output
8	1 1
1 1	2 1
2 2	3 1
3 3	
4 2	
5 3	
6 2	
7 1	
8 1	
2	
chicote 2	
vestido 3	

Problem B. Berserk

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Certa noite, após escapar de um ritual, Guts se vê amaldiçoado com uma marca no pescoço. Na noite seguinte, N monstros o cercam querendo finalizar o ritual, mas Guts resiste - ele quer derrotar o maior número possível de monstros antes de fugir com seus companheiros.

Guts consegue derrotar um monstro quando sua força é maior ou igual a do monstro. Inicialmente o i -ésimo monstro possui força A_i enquanto Guts possui força F . Guts não se importa com os ferimentos de batalha e a cada monstro derrotado dobra sua força.

Qual o maior número de monstros que Guts consegue derrotar antes de fugir?

Input

A entrada consiste de 3 linhas.

A primeira linha contém um número inteiro F , $1 \leq F \leq 2^{60}$, a força inicial de Guts.

A segunda linha contém um número inteiro N , $1 \leq N \leq 100$, a quantidade de monstros naquela noite.

A terceira linha contém N números inteiros, $1 \leq A_i \leq 2^{60}$, a força de cada monstro.

Output

Imprima o número máximo de monstros que Guts consegue derrotar.

Examples

standard input	standard output
10 2 20 1	2
100 2 10 201	1

Note

Guts pode enfrentá-los em qualquer ordem

Problem C. Coleção de Cartas

Input file: `standard input`
Output file: `standard output`
Time limit: 8 seconds
Memory limit: 256 megabytes

Erick é um rapaz que, como muitos da sua idade, gosta de jogar na Steam, uma plataforma on-line de distribuição de jogos digitais. Recentemente ele descobriu as cartas colecionáveis da Steam e que poderia ganhar dinheiro com isso.

Um jogo na Steam possui uma coleção de N cartas que podem ser obtidas comprando-se pacotes de K cartas com moedas virtuais da Steam. As cartas em cada pacote são aleatórias com distribuição uniforme.

Após completar a coleção de cartas, é possível trocar a coleção por itens que podem ser vendidos no mercado da Steam. Para saber se a compra de pacotes é lucrativa, Erick deseja saber o valor esperado da quantidade de pacotes que ele deverá comprar para completar a coleção de cartas.

O valor esperado E pode ser representado por $A\ B/C$, onde A é a parte inteira, B é o numerador da parte fracionária e C é o denominador da parte fracionária.

Input

São fornecidos dois números inteiros N e K ($1 \leq N \leq 15$ e $1 \leq K \leq \min(N, 5)$) - o número de cartas da coleção e o número de cartas por pacote respectivamente.

Output

Imprima uma única linha contendo o número A seguido do número $B * C^{-1} \pmod{10^9 + 7}$ separados por um espaço. É garantido que, com exceção dos casos de teste apresentados, é válido que $|\lfloor E \rfloor - E| > 10^{-3}$, onde $\lfloor E \rfloor$ indica o número inteiro mais próximo de E .

Examples

standard input	standard output
1 1	1 0
2 1	3 0

Note

* Baseado em fatos reais.

Problem D. Decomposição de Matrizes

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Carlos gosta muito de álgebra linear e recentemente ele viu que transformadas lineares em vetores podem ser modeladas como produtos de matrizes, ou seja, $f(v) = Av$ para uma função linear f , um vetor v e uma matriz A . Ele também percebeu que se houverem duas transformadas lineares da forma $f(v) = Av$ e $g(v) = Bv$, é possível definir uma transformada linear $h(v) = Cv$, tal que $h(v) = f(g(v))$ e $C = AB$.

Carlos ficou curioso quanto ao problema inverso, ou seja, dada uma transformada linear $h(v)$, ele deseja descobrir se é possível obter uma transformada linear $f(v)$ e uma transformada linear $g(v)$ tal que $h(v) = f(g(v))$.

Input

A primeira linha contém dois números inteiros N e M ($1 \leq N, M \leq 100$) que representam o número de linhas e colunas da matriz C . Cada uma das próximas N linhas contém M números inteiros c_{ij} ($0 \leq c_{ij} \leq 10^4$) - os elementos da matriz C .

Output

Caso seja possível obter uma solução para o problema, imprima o número de linhas da matriz A , o número de colunas da matriz A , o número de linhas da matriz B e o número de colunas da matriz B , separados por espaço. Na linha seguinte da saída, imprima a matriz A adotando o mesmo modelo de impressão da matriz C da entrada, linha a linha. Após a matriz A , imprima uma linha em branco e em seguida imprima a matriz B da mesma forma que A . Os elementos das matrizes A e B devem estar no intervalo $[0, 10^4]$. Caso não exista nenhuma solução, imprima -1 . No caso de haver múltiplas soluções para o problema, você pode imprimir qualquer uma.

Example

standard input	standard output
1 1 12	1 3 3 1 1 2 1 2 3 4

Problem E. Esferas do Dragão

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Uma terrível ameaça conseguiu aniquilar quase todos os seres vivos no Universo 7 com exceção de Goku, Vegeta (dois alienígenas superpoderosos da raça Saiyajin) e você. Após uma longa batalha, Goku e Vegeta conseguem destruir essa ameaça e agora devem juntar as sete esferas do dragão para reviver a todos que foram mortos. Contudo, eles devem ser rápidos visto que da última vez que toda a vida em um Universo foi eliminada, o deus Supremo Zeno apagou esse universo da existência.

Goku e Vegeta, equipados cada um com um Radar do Dragão (aparelho que detecta a posição de cada esfera do dragão), viajam pelo Universo 7 em busca das esferas do dragão. Ambos são igualmente rápidos, porém Goku possui a vantagem de poder se teletransportar até Vegeta a qualquer momento.

Apesar de serem muito poderosos, Goku e Vegeta não estão entre os seres mais inteligentes do Universo 7. Portanto coube a você planejar uma rota que minimize o tempo no qual eles poderão obter as sete esferas. Você deve considerar que ambos sairão da Terra (coordenada $(0, 0, 0)$) e retornarão até ela assim que terminarem a busca.

Como o radar do dragão só marca as posições das esferas e da Terra no mapa, uma restrição a ser considerada na construção da rota é que somente podem ser trilhados caminhos retilíneos com sentido para uma posição das esferas ou da Terra. Além disso, por serem impacientes, considera-se também que Vegeta e Goku só podem ficar parados na Terra.



Radar do dragão e como as esferas são apresentadas nele.

Input

Na primeira linha há um número inteiro V ($1 \leq V \leq 10^3$) que representa a velocidade dos Saiyajins. As sete próximas linhas contêm cada uma três números inteiros X_i, Y_i, Z_i ($-10^4 \leq X_i, Y_i, Z_i \leq 10^4$) representando as coordenadas tridimensionais da i -ésima esfera. A velocidade dos Saiyajins e as coordenadas tridimensionais estão na mesma unidade de comprimento.

Output

Imprima um número em ponto flutuante que representa o tempo mínimo para que Goku e Vegeta encontrem todas as esferas e retornem para a Terra para realizar seu desejo. Serão aceitas respostas cujo erro absoluto ou relativo seja menor que 10^{-4} .

Example

standard input	standard output
425 -8948 -2545 -8625 2072 2117 5945 -1464 2830 5682 -8637 3660 -9763 -4870 -6650 -147 2215 -7179 6353 -6045 -8523 -5028	68.8850

Note

A seguinte sequência de passos resulta na resposta do exemplo:

- Goku vai em direção à quinta esfera enquanto Vegeta pega a terceira esfera.
- Goku pega a quinta esfera enquanto Vegeta vai em direção à segunda esfera.
- Goku vai em direção à sétima esfera enquanto Vegeta pega a segunda esfera.
- Vegeta se dirige à sexta esfera.
- Goku pega a sétima esfera e se dirige a primeira esfera.
- Vegeta pega a sexta esfera e se encaminha para Terra.
- Goku pega a primeira esfera.
- Goku pega a quarta esfera enquanto Vegeta se dirige ao planeta Terra.
- Goku se teleporta até Vegeta e os dois vão juntos ao planeta Terra.

Problem F. Fuuton

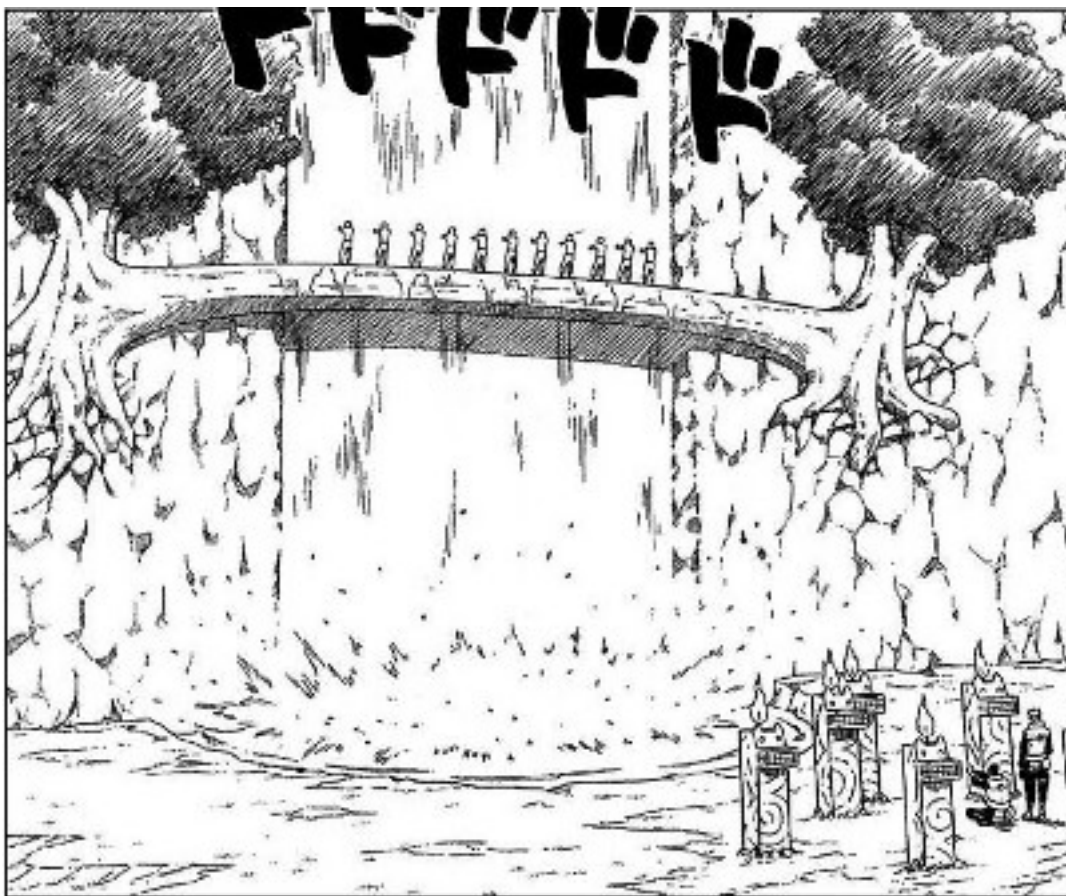
Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Naruto é um poderoso ninja da Aldeia da Folha, reconhecido pela habilidade de fazer clones de si mesmo. Cada clone tem o poder de realizar todas as ações que Naruto consegue realizar e, quando o clone é liberto, transfere ao Naruto todas as memórias e experiências pelas quais passou. Recentemente, seu professor Kakashi verificou que Naruto tem afinidade com o Fuuton (elemento vento), o que lhe permite modelar o chakra (uma mistura da energia presente em cada célula do corpo e a energia espiritual adquirida a partir de exercícios e experiência) em forma de lâminas super finas e afiadas.

Sabendo da habilidade de Naruto em realizar clones, Kakashi decidiu utilizar isso em favor do treinamento. Kakashi mandou Naruto fazer vários clones alinhados de frente a uma cachoeira. Os clones devem tentar fazer cortes na cachoeira usando o Fuuton para interromper momentaneamente seu fluxo. Durante esse treinamento, dois tipos de eventos podem acontecer:

- Um subconjunto de clones realizam um corte em um segmento da cachoeira, fazendo com que todos os clones que realizaram esse corte ganhem uma mesma quantidade de experiência.
- Naruto liberta todos os clones em um intervalo de posições, substituindo-os por novos clones com experiência nula, enquanto a experiência total dos clones libertos é absorvida por Naruto.

Ajude Kakashi a determinar a experiência acumulada de Naruto após cada ocorrência do segundo evento.



Representação do treinamento de Naruto.

Input

A primeira linha contém dois números inteiros N e Q ($1 \leq N, Q \leq 10^5$) que representam, respectivamente, o número de clones enfileirados (com posições de 1 a N) e o número de eventos. Inicialmente todos os clones possuem zero de experiência. As próximas Q linhas descrevem os eventos ocorridos, que podem ser de duas formas:

1 L R X - Houve um corte no segmento da cachoeira contido entre as posições L e R ($1 \leq L \leq R \leq N$). Todos os clones que estão nessas posições ganham X de experiência ($1 \leq X \leq 10^9$)

2 L R - Naruto libera os clones entre as posições L e R ($1 \leq L \leq R \leq N$), adquirindo a experiência de todos eles e criando novos clones nas mesmas posições com zero de experiência.

Output

Para cada evento do tipo 2, imprima uma linha contendo um número inteiro correspondendo à experiência acumulada de Naruto até o momento.

Examples

standard input	standard output
4 5 1 4 4 8 1 4 4 8 1 3 3 6 2 3 3 2 4 4	6 22
6 8 1 6 6 26 1 6 6 30 1 3 6 33 2 2 6 1 1 3 7 2 1 1 2 4 6 1 1 3 34	188 195 195

Problem G. Gotta Catch'em All

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Red é um treinador de Pokémons (monstrinhos com poderes mágicos que podem ser capturados em dispositivos eletrônicos chamados Pokébolas) que aos dez anos iniciou sua jornada para capturar todos os 151 Pokémons. No início ele não sabia muito sobre os Pokémons, porém ele acabou descobrindo algumas curiosidades sobre eles. Uma delas é que cada Pokémon possui uma *taxa de captura nominal* que é uma constante representando o quão fácil é capturá-lo. A partir dessa constante é possível calcular a *taxa de captura real*, representada por a , como segue:

$$a = \frac{(3 \cdot HP_{max} - 2 \cdot HP_{current}) \cdot rate \cdot bonus_{ball} \cdot bonus_{status}}{3 \cdot HP_{max}}$$

onde:

HP_{max} é o número máximo de hit points que aquele Pokémon tem.

$HP_{current}$ é o número atual de hit points que aquele Pokémon tem.

$rate$ é a taxa de captura nominal.

$bonus_{ball}$ é um multiplicador que depende da Pokébola utilizada (igual a 1 para *Pokeball*, igual a 1.5 para *GreatBall* e igual a 2 para *UltraBall*).

$bonus_{status}$ é um multiplicador que depende do status do Pokémon (igual a 2 para *Sleep* e *Freeze*, igual a 1.5 para *Paralyze*, *Poison* e *Burn*, e igual a 1 para *Normal*).

A partir da taxa de captura real é possível calcular a probabilidade de captura do Pokémon com a seguinte fórmula:

$$p = \frac{\lfloor a \rfloor}{255}, \text{ se } a < 255$$
$$p = 1, \text{ se } a \geq 255$$

Sua PokéAgenda é um dispositivo eletrônico que consegue obter as grandezas necessárias de cada Pokémon, mas não lhe diz qual a probabilidade de captura. Sua tarefa é calcular a probabilidade de capturar cada um dos Pokémons que faltam para Red.

Input

A primeira linha contém um número inteiro N ($1 \leq N \leq 151$) que indica o número de Pokémons que faltam para Red. Cada uma das próximas N linhas inicialmente contém três strings S , T e B ($1 \leq |S|, |T|, |B| \leq 10$) que representam, respectivamente, o nome do Pokémon (não haverá Pokémons repetidos), o status do Pokémon e a bola utilizada (o nome do status e da bola terão a mesma grafia conforme descrição do problema). Continuando na mesma linha seguem quatro números inteiros que são ind (o identificador do Pokémon na lista da Pokédex), HP_{max} , $HP_{current}$ e $rate$ ($1 \leq ind \leq 151$, $1 \leq HP_{max}, HP_{current} \leq 714$, $1 \leq rate \leq 255$). Todas as strings da entrada serão compostas apenas por letras do alfabeto.

Output

Imprima N linhas em ordem crescente de ind , onde em cada linha deve ser impresso o inteiro ind , a string S e um número em ponto flutuante representando o valor de p correspondente à probabilidade de captura do Pokémon (será tolerado um erro absoluto ou relativo de até 10^{-4}).

Examples

standard input	standard output
5 Omastar Freeze GreatBall 139 178 19 45 Graveler Sleep UltraBall 75 225 199 120 Venonat Poison GreatBall 48 243 193 190 Bulbasaur Burn PokeBall 1 153 2 45 Gloom Normal GreatBall 44 191 37 120	1 Bulbasaur 0.25882 44 Gloom 0.61176 48 Venonat 0.78824 75 Graveler 0.76863 139 Omastar 0.49020
2 Arbok Freeze PokeBall 24 244 242 90 Paras Sleep GreatBall 46 193 55 190	24 Arbok 0.23529 46 Paras 1.00000

Note

$\lfloor a \rfloor$ significa arredondar o número a para baixo

Problem H. Honorável Espadachim

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

No vilarejo Rulid uma tarefa tem sido passada ao longo de gerações. Trata-se da tarefa de cortar a árvore gigante *Gigas Cedar*, atualmente passada para Eugeo que tem tentado concluir essa tarefa pelos últimos seis anos. Certo dia, surgiu diante dele o verdadeiro e único Black Swordsman, Kirito, que ensinou a Eugeo habilidades de espada e como usar a Blue Rose Sword. Eugeo acredita que agora está apto a cortar a árvore gigante.

Cada golpe de espada que Eugeo desfere provoca X de dano na árvore. Além disso, a cada K golpes de espada, Eugeo melhora sua habilidade e seu dano é acrescido em Y . Sabendo que *Gigas Cedar* tem H pontos de vida, descubra quantos golpes de espada Eugeo deverá desferir até derrubar a árvore.



O vilarejo Rulid e a árvore gigante *Gigas Cedar*.

Input

A entrada é composta por uma linha contendo quatro inteiros X , K , Y e H ($1 \leq X, Y, K \leq 1000, 1 \leq H \leq 10^{18}$).

Output

Imprima o número de golpes de espada que Eugeo deverá desferir até derrubar a árvore.

Examples

standard input	standard output
7 2 4 10	2
3 2 3 14	4
3 8 3 12	4

Problem I. Idioma Digital

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Izzy é uma das crianças escolhidas para entrar no Digimundo (o mundo dos “Digimons”, monstros digitais) tendo como parceiro Digimon o Tentomon. Izzy é muito curioso e inteligente e ficou fascinado ao chegar no Digimundo, pois viu que haviam muitas semelhanças entre o mundo digital e os programas de computadores. A escrita utilizada no Digimundo usa um código binário muito parecido com o código ASCII (veja tabela abaixo). Contudo, no Digimundo os bits estão todos trocados e são apresentados na ordem inversa. Por exemplo, o caractere ‘M’ tem valor decimal 77 na tabela ASCII que em binário é 1001101. No Digimundo os bits são trocados (0110010) e em seguida apresentados em ordem reversa (0100110). Quando voltou do Digimundo, Izzy passou um desafio para você: faça um programa que realiza a tradução de uma frase para o idioma digital do Digimundo.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[59	3B	;	91	5B	[123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-~	63	3F	?	95	5F	_	127	7F	DEL

Input

A entrada é composta de uma linha que contém uma frase a ser traduzida para o idioma digital do Digimundo. A frase é uma string com até 10^5 caracteres que na tabela ASCII estão entre o caractere Space (valor decimal 32) e o caractere ~ (valor decimal 126).

Output

Imprima em uma linha os $7 \times |S|$ caracteres que correspondem a representação digital da frase no Digimon.

Examples

standard input	standard output
Tai	110101001111000110100
Izzy	0110110101000010100000110000
Sora	0011010000010010110000111100

Problem J. Jornada para Laftel

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Riqueza, fama, poder. O homem que conseguiu tudo neste mundo, o rei dos piratas, Gol D. Roger. As últimas palavras ditas antes de sua execução, enviadas para todo o mar:

“Minhas riquezas e tesouros? Se vocês quiserem, eu os deixo pegar. Procurem por ele, deixei tudo naquele lugar!”

Os homens agora correm em direção à Grande Linha em busca de seus sonhos. O mundo entra na grande Era dos Piratas!

Luffy é mais um desses homens que busca os seus sonhos e deseja encontrar o tesouro deixado por Gol D. Roger e ter uma incrível aventura. Durante sua jornada ao longo da Grande Linha, ele descobriu que o tesouro encontra-se na ilha de Laftel. A localização de Laftel pode ser determinada a partir das posições dos quatro Road Poneglyphs. Cada Road Poneglyph encontra-se em uma coordenada no mapa e o ponto de cruzamento entre as diagonais do quadrilátero formado por eles indica a localização de Laftel.



Exemplo das quatro posições dos Road Poneglyphs.

Input

São fornecidas quatro linhas, cada uma contendo dois números inteiros x, y ($-5000 \leq x, y \leq 5000$) representando a coordenada indicada por um dos Road Poneglyphs. Garante-se que os quatro pontos estão em posição geral, ou seja, tomando-se quaisquer três pontos garante-se que eles não serão colineares e que o quadrilátero resultante é convexo.

Output

Imprima uma única linha contendo dois números em ponto flutuante representando as coordenadas de Laftel. Serão aceitas respostas cujo erro absoluto ou relativo seja menor que 10^{-6} .

Examples

standard input	standard output
0 0 -1 1 1 1 0 2	0 1
0 0 0 1 1 0 1 1	0.5 0.5

Note

Considere apenas nesta questão que a Terra é plana.

Problem K. Kiyo e seus primos

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

Kiyo é um garoto de 14 anos muito inteligente que está realizando um trabalho escolar sobre a conjectura de Goldbach. Essa conjectura diz que todo número par maior que 2 pode ser representado pela soma de dois números primos. Para seu estudo, Kiyo montou uma pilha de papéis com números primos que serviam para consulta. No entanto seu amigo, Gash, acabou deixando alguns desses papéis voarem pela janela. Quando Kiyo percebeu isso, ficou irritado, porém viu uma oportunidade de resolver um outro problema interessante. Ele agora quer saber quantos pares de números primos (dentre aqueles que não voaram pela janela) tem a soma igual a certos números inteiros.

Input

A primeira linha contém um número inteiro N ($0 \leq N \leq 78498$) que é a quantidade de números que Gash deixou escapar. Em seguida os N números primos perdidos são listados, um por linha. Nenhum primo excede o valor de 10^6 . Em seguida haverá uma linha com um número inteiro Q ($1 \leq Q \leq 10^6$) que é o número de consultas que Kiyo fará. Q linhas seguem contendo em cada uma um número b_j ($1 \leq b_j \leq 10^6$) que deve ser consultado.

Output

Para cada consulta imprima uma linha contendo a quantidade de pares (não-ordenados) de números primos cuja soma é igual ao valor consultado. Obs: considera-se, por exemplo, que os pares não-ordenados 3, 7 e 7, 3 são iguais.

Examples

standard input	standard output
0	1
5	1
4	1
6	2
8	1
10	
12	
1	1
3	0
4	0
4	1
5	
6	
12	