G César64

Limite de Tempo: 1s

Após um seminário sobre criptografia, o jovem César resolver exercitar seus conhecimentos recém-adquiridos propondo uma nova forma de criptografia simétrica, baseada na cifra de César clássica, e a denominou "César64".

Esta nova cifra funciona da seguinte forma:

1. O emissor escolhe uma mensagem M, composta apenas por caracteres ASCII imprimíveis (isto é, letras maísculas e minúsculas, dígitos decimais, pontuações e o espaço em branco);
2. Em seguida, o emissor preenche um vetor v, onde cada entrada $v[i]$ tem dois *bytes* de tamanho e contém o inteiro que corresponde ao caractere $M[i]$ na tabela ASCII;
3. O emissor escolhe uma chave secreta K, com $1 \leq K \leq 65535$;
4. O emissor cria um novo vetor w, a partir de v, onde $w[i] = (v[i] + K) (\text{mod } 2^{16})$. Assim como v, cada elemento $w[i]$ de w ocupa exatamente 2 *bytes*;
5. Por fim, os *bytes* contidos em w são codificados em base 64.

O resultado final C do procedimento anterior é então enviado para o destinatário que, conhecendo a chave K de antemão, decodifica C, obtem o vetor v a partir de w e, em seguida, a mensagem M.

César estava convencido que sua nova variante era superior, em termos de segurança, do que o algoritmo clássico, até que seu amigo Nero lhe disse "Esta nova forma é tão vulnerável quanto à anterior a um ataque de mensagem conhecida". Mostre que Nero está certo: dada uma mensagem criptografada C, e um texto T que estava presente na mensagem original M (isto é, T é substring de M), recupere a chave K.

Base 64

O termo Base 64 se refere a um grupo de esquemas de codificação para a representação de dados binários em uma string ASCII, onde os símbolos são mapeados em um sistema de numeração de base 64.

O esquema de codificação consiste em representar cada grupo de 6 bits por um caractere, segundo a figura abaixo.

Se o conjunto de bytes a serem codificados não for um múltiplo de 3, então são acrescidos, ao final do conjunto, um ou dois bytes, com o valor zero, para que o total se torne então múltiplo de três. Caso o resto da divisão do número original de bytes por 3 for igual a 1, são codificados apenas os 12 primeiros bits do últmo bloco; se o resto for 2, são codificados apenas os 18 primeiros bits do bloco final.

Para indicar que o último grupo consiste de apenas 1 ou 2 bytes, serão anexados, ao final da codificação, os caracteres "=", ou "==", respectivamente.

Entrada

Valor	Caractere	Valor	Caractere	Valor	Caractere	Valor	Caractere
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

A entrada consiste em, no máximo, 30 casos de teste. Cada caso de teste é composto por duas linhas: a primeira contém uma string T ($2 \leq |T| \leq |M|$), que corresponde à uma substring da mensagem original M ($1 \leq |M| \leq 3 \times 10^5$); a segunda contém a mensagem criptografada C ($1 \leq |C| \leq 10^6$). A string T é composta apenas por caracateres ASCII imprimíveis e a string C contém apenas caracateres válidos da codificação base 64.

Saída

Para cada caso de teste imprima, em uma linha, a mensagem "Caso #t: K", onde t é o número do caso de teste (cuja contagem inicia com o número um) e K é a chave utilizada para se obter C a partir de M. Se houver mais de uma chave possível, escolha a menor delas.

Exemplos de entradas	Exemplos de saídas
dia uFG4frh8uC+4c7h4uHC4MA== UnB Xste5F64XpZevV7XXuNe1w== de X/tgHWAaYCFgDF/LYA9gEF/LX/9f8F/7 2/2 iYiJiomHiYqJiImJiY8=	Caso #1: 47119 Caso #2: 24182 Caso #3: 24491 Caso #4: 35160

Este problema foi elaborado para ensino e docência. Quaisquer coincidências com problemas já existentes favor entrar em contato (edsonalves@unb.br) para que as devidas providências sejam tomadas.