MARATONA DE PROGRAMAÇÃO
InterFatecs

Problema I

Criptografia hash

Arquivo fonte: criptografia.{ c | cpp | java | py }

Autor: Prof. Me. Lucio Nunes de Lira (Fatec Diadema e Fatec Ferraz de Vasconcelos)

Daiane, profissional formada em arquitetura, gosta muito de matemática, tecnologia e segurança da informação, tanto que está cogitando seriamente ingressar em um curso superior na área de ciências da computação. Nesta área, o que mais chamou sua atenção foi a possibilidade em trabalhar com programação de computadores.

Ao conversar com alguns amigos programadores, percebeu que a habilidade de programar computadores está cada vez mais requisitada e que quanto maior a familiaridade com matemática, maior a facilidade para programar e vice-versa, inclusive escutou diversas frases do tipo "[...] eu nem gostava muito de matemática, mas quando comecei a programar entendi a aplicação daquilo que era dado na escola, como a 'Fórmula de Bhaskara' para encontrar as raízes de uma equação de segundo grau, o 'Teorema de Pitágoras' sobre os triângulos retângulos e a importância dos números primos".

A respeito dos números naturais primos, é fácil lembrar do que se tratam: são números naturais maiores do que 1 e que possuem apenas dois divisores naturais, o 1 e ele próprio. Daiane também sabe que os números primos são usados em algoritmos para criptografar dados, isto é, embaralhar uma mensagem e, consequentemente, reduzir a probabilidade de alguém não autorizado ter acesso ao seu conteúdo original, o que é essencial para manter a privacidade e a segurança desses dados.

Existem diversos algoritmos para criptografia, alguns permitem que os dados sejam embaralhados (cifragem) e, posteriormente desembaralhados (decifragem), algo útil na transmissão de mensagens entre pessoas. Outros algoritmos permitem apenas a cifragem, tornando quase impossível que o valor gerado a partir desse embaralhamento seja usado para retornar ao conteúdo original. Esses algoritmos são bastante usados para guardar senhas em bancos de dados. Como exemplo de aplicação dos algoritmos que permitem apenas cifragem, vamos supor que um cliente quer criar um usuário com senha para ter acesso a um sistema. Podemos resumir o procedimento assim:

- O cliente insere em uma tela de cadastro um nome de usuário e senha;
- O programa gera um valor com base em um embaralhamento desta senha, geralmente chamado de "valor hash", e guarda no banco de dados apenas o nome de usuário e esse valor hash;
- Para acessar o sistema, o cliente irá inserir seu nome de usuário e a senha;
- O sistema embaralhará a senha inserida e irá comparar esse valor de hash com aquele guardado no banco de dados e que está associado ao mesmo nome de usuário;
- Se os valores forem iguais, o acesso será concedido, caso contrário, o acesso será negado. Afinal, o algoritmo garante que o valor de hash será sempre o mesmo para os mesmos dados.

Essa estratégia é segura, pois caso o banco de dados seja acessado indevidamente, as senhas originais dos usuários não estarão expostas, apenas o valor hash. Que bom seria se todas as empresas tivessem esse cuidado com os dados de seus clientes...

Mal Daiane anunciou que pretende mudar de área e já recebeu um convite de sua amiga Michelly, dona de uma pequena empresa. A empresária disse que precisa melhorar a segurança de seu sistema e que pretende

coletar todas as senhas de seus clientes e guardá-las usando a estratégia que vimos anteriormente. Daiane então elaborou um algoritmo para criptografar as senhas:

1. Coleta-se o nome de usuário e a senha de um cliente;
2. Converte-se cada caractere da senha para o seu respectivo código ASCII (vide Figura I.1);
3. Multiplica-se cada código pelo número de sua posição, sendo que o primeiro está na posição 1, o segundo está na posição 2 e assim por diante;
4. Somam-se todos os códigos já multiplicados por suas posições, gerando um número natural S;
5. Calcula-se a decomposição de S em fatores primos, isto é: (a) cria-se uma lista vazia L; (b) encontra-se o menor número primo P que é divisor de S; (c) guarda-se P no final de L; (d) encontra-se o quociente Q da divisão de S por P; (e) atribui-se a S o valor de Q; (e) repete-se os passos (b), (c), (d) e (e) até que S seja 1;
6. Em L estão todos os fatores primos que, se multiplicados, resultam no S original, ou seja, o valor de S antes de ser alterado por consequência do passo 5. Note que, por razão do procedimento, podem existir fatores repetidos em L, o que aconteceria, por exemplo, se S fosse 100 (2 * 2 * 5 * 5);
7. Transforma-se todo fator primo de L em um texto, concatenando o fator que estiver na posição i+1 à direita do fator que estiver na posição i, gerando um único texto F;
8. Concatena-se o texto correspondente ao conteúdo do S original à esquerda de F, gerando um texto H, esse é o valor de hash do algoritmo de Daiane. Veja a ilustração de exemplo na Figura I.2.

Caracteres e respectivos códigos ASCII (tabela parcial)

Caractere	Código	Caractere	Código	Caractere	Código	Caractere	Código	Caractere	Código
A	65	N	78	a	97	n	110	0	48
B	66	O	79	b	98	o	111	1	49
C	67	P	80	c	99	p	112	2	50
D	68	Q	81	d	100	q	113	3	51
E	69	R	82	e	101	r	114	4	52
F	70	S	83	f	102	s	115	5	53
G	71	T	84	g	103	t	116	6	54
H	72	U	85	h	104	u	117	7	55
I	73	V	86	i	105	v	118	8	56
J	74	W	87	j	106	w	119	9	57
K	75	X	88	k	107	x	120		
L	76	Y	89	l	108	y	121		
M	77	Z	90	m	109	z	122		

Figura I.1: Tabela ASCII parcial, apenas com os caracteres e códigos usados no problema.

Você, que tem bastante experiência com programação, se ofereceu para ajudar sua amiga Daiane a implementar esse algoritmo em uma linguagem de programação. Para isso seu programa irá receber diversas entradas com o nome de usuário e senha de clientes e, para cada cliente, irá exibir o correspondente nome de usuário e o valor de hash da senha, conforme procedimento criado por ela. Atenção as saídas devem estar ordenadas alfabeticamente pelo nome de usuário.

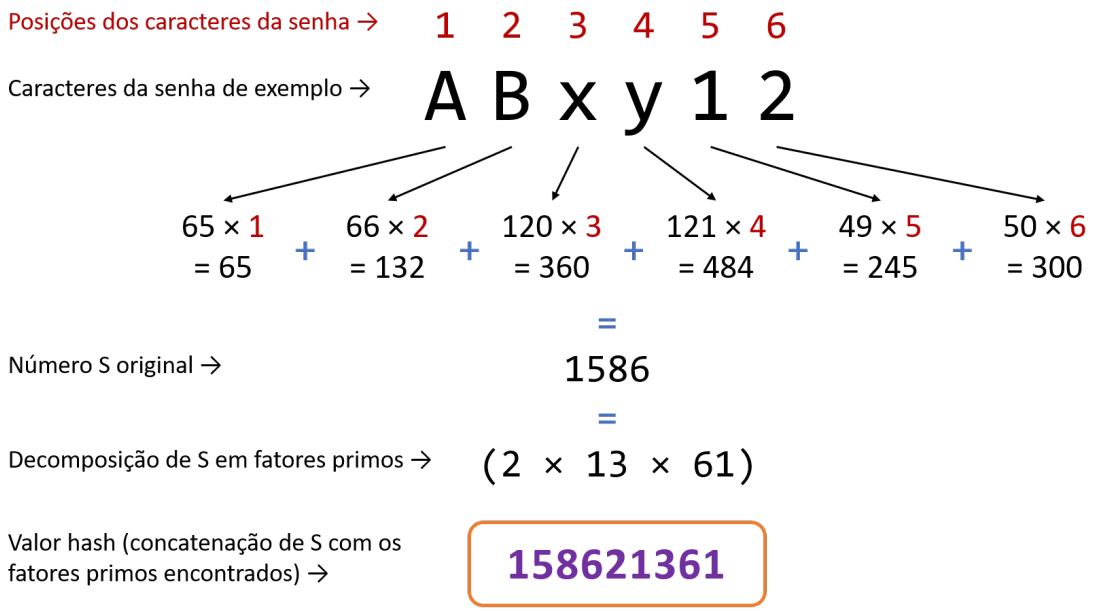

Figura I.2: Ilustração do procedimento para geração do valor hash do algoritmo de Daiane.

Entrada

A entrada é composta por uma quantidade variável de linhas L $(1 \leqslant L \leqslant 2000)$, em que cada linha contém um nome de usuário U e uma senha S. No sistema da empresa de Michelly, U é o próprio nome do cliente, porém, sem espaços e compostos apenas por letras minúsculas e não acentuadas do alfabeto latino (a..z), limitado a 50 caracteres. O sistema de Michelly autoriza senhas com no máximo 100 caracteres, por isso S é composto apenas de letras maiúsculas não acentuadas do alfabeto latino (A..Z), letras minúsculas e não acentuadas do alfabeto latino (a..z) e dígitos decimais (0..9). A entrada é encerrada com a palavra *ACABOU*, com todos os caracteres maiúsculos.

Saída

A saída deverá ser o nome de usuário em uma linha e, na linha seguinte, o respectivo valor hash calculado com base na senha associada ao mesmo usuário. Após cada par de linhas devem existir trinta caracteres '-' (hífen), sem os apóstrofos, seguido de uma quebra de linha, vide exemplos. Note que os usuários deverão ser exibidos ordenados alfabeticamente pelo nome de usuário.

Exemplo de Entrada 1	**Exemplo de Saída 1**

`megandenise ABxy12`
`ACABOU`
	`usuario...: megandenise`
	`valor hash: 158621361`
	`------------------------------`

Exemplo de Entrada 2	Exemplo de Saída 2
clara ABC123 ana abc123 bia a1b2c3 duda 1a2b3c andrecavalcante ZZZzzz zequinha 1234567890 helena x9x9 ACABOU	usuario...: ana valor hash: 134221161 ------------------------------ usuario...: andrecavalcante valor hash: 237023579 ------------------------------ usuario...: bia valor hash: 149025149 ------------------------------ usuario...: clara valor hash: 115025523 ------------------------------ usuario...: duda valor hash: 163421943 ------------------------------ usuario...: helena valor hash: 82223137 ------------------------------ usuario...: zequinha valor hash: 2925335513 ------------------------------

Exemplo de Entrada 3	Exemplo de Saída 3
jaentendeualogica QUEROSOHVER ACABOU	usuario...: jaentendeualogica valor hash: 5174213199 ------------------------------

Exemplo de Entrada 4	Exemplo de Saída 4
maria 2AB ACABOU	usuario...: maria valor hash: 37823337 ------------------------------