Algoritmos e Programação de Computadores
Projeto 04: Cifra de Vigenère
ABI/LFI/TAI

Prof. Daniel Saad Nogueira Nunes

1 Contextualização

A Cifra de Vigenère é uma cifra de substituição polialfabética originalmente descrita por Giovan Battista Bellaso em 1553 em seu livro *La cifra del. Sig. Giovan Battista Bellaso*. Contudo, ela foi posteriormente e erroneamente atribuída à Blaise de Vigenère no século XIX.

Apesar da cifra ser facilmente entendível, ela resistiu a ataques por quase 3 séculos, o que rendeu a ela durante este período o título *le chiffre indéchiffrable*.

Esta cifra baseia-se em sucessivas aplicações de Cifras de César. Em uma Cifra de César, cada símbolo do texto é rotacionado de um número fixo de posições. Por exemplo, caso a cifra rotacione os caractere em 3 posições e o alfabeto seja composto das letras $\Sigma = \{A, B, \ldots, Z\}$, a cifra possuirá as seguintes substituições:

$$\begin{aligned} A &\mapsto D \\ B &\mapsto E \\ C &\mapsto F \\ &\vdots \\ X &\mapsto A \\ Y &\mapsto B \\ Z &\mapsto C \end{aligned}$$

As substituições efetuadas pelas cifra também pode ser conferidas na Figura 1.

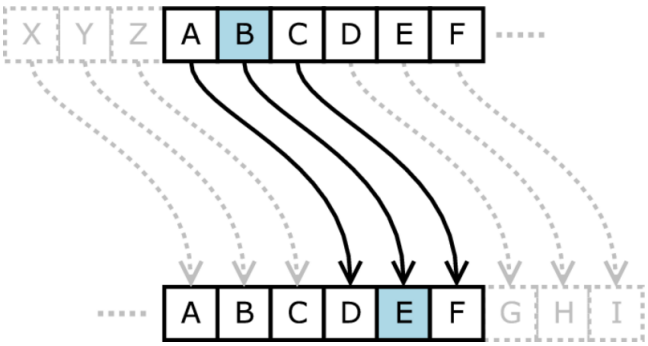

Figura 1: Cifra de César.

Uma cifra de Vigenère utiliza uma chave k com símbolos do alfabeto Σ para cifrar um texto T, também sobre o alfabeto Σ. Cada símbolo da chave indica quantas rotações devem ser feitas em cada símbolo do texto. Assim, se o alfabeto for $\Sigma = \{A, B, \ldots, Z\}$, o símbolo A na chave indica que o símbolo correspondente do texto deve ser rotacionado de 0 posições, o símbolo B especifica que o símbolo correspondente no texto deve ser rotacionado de 1 posição, e assim por diante. Para efetuar a substituição, a chave é colocada sobre o texto e o símbolo do texto é rotacionado de acordo com o símbolo correspondente da chave.

Se o texto corresponde à T = "ABRACADABRA"e a chave k = "MAGICA", a cifra de Vigenère ao aplicar a chave no texto corresponde ao indicado pela Tabela 1.

Note que, se a chave é menor que o texto, ela deve ser aplicada diversas vezes de maneira cíclica.

Tabela 1: Exemplo da aplicação de uma Cifra de César.

Texto	`ABRACADABRA`
Chave	`MAGICAMAGIC`
Texto Cifrado	`MBXIEAPAHZC`

2 Especificação

O objetivo deste projeto é a implementação da cifração e decifração da Cifra de Vigenère.

O alfabeto Σ a ser considerado possui as letras maiúsculas, minúsculas e símbolos especiais. Em resumo, são todos os símbolos no intervalo $[33, 126]$ da tabela ASCII, ilustrada pela Figura 2.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Figura 2: Tabela ASCII.

Os espaços no texto e caracteres de nova linha não devem ser cifrados, uma vez que não fazem parte do alfabeto Σ.

Uma vez que a chave é menor que o texto, ela deve ser usada múltiplas vezes, até cifrar/decifrar o texto inteiro. No caso do texto ter múltiplas linhas, o método de cifração/decifração continua usando a chave de onde parou na linha imediatamente anterior.

2.1 Entrada e Saída

Os dados de entrada deverão ser lidos do teclado `stdin` e os dados de saída deverão ser escritos na tela (`stdout`). É muito importante que o programa siga rigorosamente o formato de entrada e saída. Nada além do que está especificado deve ser impresso em tela. Não é necessário validar a entrada, é garantido que as entradas utilizadas para testar o programa seguem a especificação.

2.2 Entrada

A entrada consiste de **pelo menos** três linhas. A primeira linha corresponde a um inteiro que pode assumir os valores 0 ou 1. Se o inteiro é 0, então o objetivo é a cifração. Se o inteiro é 1, o objetivo é decifração.

A segunda linha corresponde à chave. Esta linha não possui espaços e é composta apenas por símbolos do alfabeto Σ.

As próximas linhas correspondem ao texto, que deve ser cifrado ou decifrado, de acordo com o inteiro e a chave lidos. Note que o texto pode ter várias linhas.

Tanto a chave quanto cada linha do texto não excedem 80 caracteres. Assuma que o inteiro só poderá assumir os valores 0 e 1.

2.3 Saída

A saída do programa deve ser somente o texto cifrado/decifrado de acordo com a opção de cifração/decifração, a chave dada, e o texto original/cifrado, conservando os espaços originais e os caracteres de nova linha. Nada além do mencionado deve ser impresso pelo seu programa.

2.3.1 Exemplos de cifração

Entrada	Saída
0 MAGICA ABRACADABRA	mbxieapahzc

Entrada	Saída
0 MAGICA ABRA CA DABRA PE DE CABRA	mbxi ea pahzc pq dk kcb~a

Entrada	Saída
0 Teste@Chave O rato roeu a roupa do rei de roma! Mas o rato nao morreu! Dona xica, admirou-se! Do berro que o rato deu!	$ XUiU 33NW X XD[dV J0 6NK [K GUaVe l%\ Q iGIU bVU .3[T\[T *ccG 9-LC# G9S]gU6O\Gv *D HYgX0 5^G f X6Zc YK6C

2.3.2 Exemplos de decifração

Entrada	Saída
1 Ch@veDo!d4 uR. jG1QbN"P 10Z[8{pTv9\@ 8\&Ta 0x(J7iG^	Sim salabim. Hocus-pocus! Avada Kedavra;

Entrada	Saída
1 chavetetra ' RU eUi G W\I PJP^[VMZ} CRM Kk Oh Tdg YIUN jKg\ZW D] J D‘M WUd^n ot $iOVT L! -I[P‘M]Gc} 6LN % GXdMgTOQRP^ 2VT\hCKN	C is not a big language, and it is not well served by a big book. -- Brian W. Kernighan, The C Programming Language

2.4 Documentação

O código deverá ser devidamente indentado, documentado através de comentários e respeitando as boas práticas de programação considerando os nomes das variáveis, caso contrário, haverá desconto na nota do projeto.

Além disso, no cabeçalho do arquivo fonte deverá constar o nome e matrícula do aluno.

2.5 Ambiente de Testes

A especificação do ambiente que será utilizado para testes e correção dos trabalhos segue abaixo:

- Sistema: `Manjaro GNU/Linux Kernel 5.10.42-1`;
- Compilador: `gcc 11.1.0`;

3 Considerações

- Este projeto deve ser executado **individualmente**.
- A incidência de plágio acarretará automaticamente nota 0 (**zero**) para os envolvidos. Medidas disciplinares também serão tomadas.
- Trabalhos que não compilarem não serão avaliados.
- O código deve ser entregue em uma pasta zipada com a devida identificação do aluno através da sala de aula virtual da disciplina.