



UNISANTOS

Universidade Católica de Santos

Centro de Ciências Exatas, Arquitetura e Engenharia

Professor:	Ciro Cirne Trindade
Disciplina:	Introdução à Computação-II
Cursos:	Ciência da Computação/Sistemas de Informação
Data de entrega:	25/09/2018
Grupos:	3 alunos

1º Trabalho Prático – Enigma¹

Dada uma configuração inicial, a máquina de criptografia alemã Enigma, da Segunda Guerra Mundial, substituía cada letra digitada no teclado por alguma outra letra. A substituição era bastante complexa, mas a máquina tinha uma vulnerabilidade: uma letra nunca seria substituída por ela mesma! Essa vulnerabilidade foi explorada por Alan Turing, que trabalhou na criptoanálise da Enigma durante a guerra. O objetivo era encontrar a configuração inicial da máquina usando a suposição de que a mensagem continha uma certa expressão usual da comunicação, como por exemplo a palavra ARMADA. Essas expressões eram chamadas de *cribs*. Se a mensagem cifrada era, por exemplo, FDMLCRDMRALF, o trabalho de testar as possíveis configurações da máquina era simplificado porque a palavra ARMADA, se estivesse nessa mensagem cifrada, só poderia estar em duas posições, ilustradas na tabela abaixo com uma seta. As demais cinco posições não poderiam corresponder ao *crib* ARMADA porque ao menos uma letra do *crib*, sublinhada na tabela abaixo, casa com sua correspondente na mensagem cifrada; como a Enigma nunca substituiria uma letra por ela própria, essas cinco posições poderiam ser descartadas nos testes.

F	D	M	L	C	R	D	M	R	A	L	F
A	R	<u>M</u>	A	D	A						
	A	R	M	A	D	A	←				
		A	R	M	A	<u>D</u>	A				
			A	R	M	A	D	A	←		
				A	<u>R</u>	M	A	D	<u>A</u>		
					A	R	<u>M</u>	A	D	A	
						A	R	M	<u>A</u>	D	A

Neste problema, dada uma mensagem cifrada e um *crib*, seu programa em C deve computar o número de posições possíveis para o *crib* na mensagem cifrada.

Entrada

A primeira linha da entrada contém a mensagem cifrada, que é uma sequência de, pelo menos, uma letra e no máximo 10^4 letras. A segunda linha da entrada contém o *crib*, que é uma sequência de, pelo menos, uma letra e no máximo o mesmo número de letras da mensagem. Apenas as 26 letras maiúsculas, sem acentuação, aparecem na mensagem e no *crib*.

¹ Extraído e adaptado do problema E (Enigma) da prova da 1ª fase da Maratona de Programação da SBC de 2018.



Saída

Imprima uma linha contendo um inteiro, indicando o número de posições possíveis para o *crib* na mensagem cifrada.

Exemplo

Entrada	Saída
FDMLCRDMRALF ARMADA	2

Entrada	Saída
AAAAABABABABABABABABA ABA	7

Para resolver este problema você deve obrigatoriamente implementar a seguinte função:

```
/* devolve verdadeiro se é possível que o crib esteja localizado  
   a partir da posição index da mensagem */  
bool eh_possivel(int index, const char * mensagem, const char * crib);
```

Informações importantes sobre o trabalho

1. Critérios de avaliação:
 - a) Corretude: 90%
 - b) Legibilidade: 10%
2. Todos os trabalhos devem possuir no cabeçalho a identificação dos autores.
3. Trabalhos copiados (com ou sem eventuais disfarces) terão nota dividida pelo número de cópias (inclusive o original).
4. Trabalhos atrasados terão um desconto de 1 ponto por dia atrasado, até o limite de 5 dias de atraso.
5. Trabalhos com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO.
6. É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa. A legibilidade do código será levada em consideração pelo critério de avaliação do trabalho.
7. Você deve enviar seu arquivo fonte (.c) da sua solução através do link específico no Moodle.