



UNIVERSIDADE ESTADUAL PAULISTA
“Julio de Mesquita Filho”
FACULDADE DE ENGENHARIA

6ª. LISTA DE EXERCÍCIOS – STRINGS

Disciplina: Programação de Computadores

Prof. Dra Cassilda Maria Ribeiro

Prof. Dr. Anibal Teixeira

Ministrada por: prof. André Amarante Luiz

Exercício 1: Uma palavra é denominada um palíndromo se for invertida e a leitura da mesma permanecer sem nenhuma alteração. Algumas palavras que são palíndromos são: aba, radar, reter, rever, rir, rotor, dentre outras. Construir um programa que detecte se uma palavra (string) digitada pelo usuário é ou não um palíndromo.

Exercício 2: Existem palíndromos que são formados por frases tais como:

(i) Socorram-me subi no onibus em Marrocos.

(ii) Omitiram radar maritmo.

Para mais exemplos consulte o seguinte endereço da internet:

<http://pt.wikipedia.org/wiki/Pal%C3%ADndromo>.

Construir um programa que ignore os espaços e o caractere '-' ao verificar se uma frase é um palíndromo.

Exercício 3: Construir um programa que conta e imprime o número de ocorrências de uma letra, fornecida pelo usuário, em uma dada string, também digitada pelo usuário.

Exercício 4: Construir um programa que pede para o usuário:

(i) Uma string s,

(ii) Um caractere ch1,

(iii) Um caractere ch2.

O programa deve substituir todas as ocorrências do caractere ch1 em s pelo caractere ch2.

Exercício 5: Construir um programa que armazena em um vetor de inteiros v os índices de onde um caractere ch1 digitado pelo usuário aparece em uma string s também digitada pelo usuário. Depois de obter o vetor v, o mesmo deve ser impresso. Um exemplo de execução do programa é dado por:

Digite uma string: Uma longa jornada.

Digite o caractere: a

Índices onde ocorrem o caractere a: 2 8 14 16

Para verificar que o exemplo de execução está correto, basta ver que:

S=	U	m	a		l	o	n	g	a		j	o	r	n	a	d	a	.
i=	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
			↓						↓						↓		↓	
V=	[2 8 14 16]																	

Exercício 6: Construir um programa que leia uma string s1 e uma substring s2 e depois fornece o número de ocorrências da substring s2 em uma string s1. Um exemplo de resposta que o programa deve fornecer é dado abaixo:

Digite a string: O rato roeu a roupa do rei de Roma e a rainha de raiva roeu o rato.
 Digite a substring: ra
 Numero de ocorrências: 4

S=	O		r	a	t	o		r	o	e	u		a		r	o	u	p	a
i=	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
			↓																
			1																

S=		d	o		r	e	i		d	e		r	o	m	a		e		
i=	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		

S=	a		r	a	i	n	h	a		d	e		r	a	i	v	a		
i=	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	52	53		
			↓										↓						
			2										3						

S=	r	o	e	u		o		r	a	t	o	.
i=	54	55	56	57	58	59	60	61	62	63	64	65
								↓				
								4				

Assim, no exemplo anterior o número de ocorrências é realmente igual a 4.

Exercício 8: Generalize o **Exercício 7** para que seu programa seja capaz de ignorar diferenças entre maiúsculas ou minúsculas digitadas na string ou na substring. Por exemplo:

Digite a string: O rato roeu a roupa do rei de Roma e a rainha de raiva roeu o rato.
 Digite a substring: RA
 Numero de ocorrências: 4

Exercício 9: Construir um programa que seja capaz de substituir ocorrências de uma substring sub1 em uma string s por outra substring sub2 e mostra a nova string s. Um exemplo é:

Digite a string: O rato roeu a roupa do rei de Roma e a rainha de raiva roeu o rato.

Digite a substring sub1: ra
 Digite a substring sub2: pa
 Digite a string: O pato roeu a roupa do rei de Roma e a painha de paiva roeu o pato.

Exercício 10: Construir um programa que seja capaz de concatenar uma string s1 e uma outra string s2 em uma string s3. Por exemplo:

Digite a string s1: Quem canta os males espanta.
 Digite a string s2: Ha males que vem para o bem.
 Nova string s3: Quem canta os males espanta. Ha males que vem para o bem.

Exercício 11: Construir um programa que seja capaz de concatenar uma string s1 e uma outra string s2 na própria string s2. Por exemplo:

Digite a string s1: Quem canta os males espanta.
 Digite a string s2: Ha males que vem para o bem.
 Nova string s2: Ha males que vem para o bem. Quem canta os males espanta.

Exercício 12: Construir um programa que seja capaz de embaralhar uma string s1 com uma string s2 e colocar o resultado em uma string s3. Para embaralhar S1 com S2 é necessário preencher os índices pares de S3 com os elementos de S1 e os ímpares com os elementos de S2 até que os elementos de uma das duas strings termine e os demais elementos de S3 serão preenchidos com os elementos da string restante. Por exemplo:

Digite a string s1: local.
 Digite a string s2: misterio.
 Nova string s3: lmoicsatlerio.

Esquemáticamente:

S1=	l		o		c		a		l				
	↓	→	↓	→	↓	→	↓	→	↓	→			
S2=	m		i		s		t		e		r	i	o
S3=	l	m	o	i	c	s	a	t	l	e	r	i	o
	0	1	2	3	4	5	6	7	8	9	10	11	12

Exercício 13: Construir um programa que seja capaz de embaralhar uma string s1 com uma string s2 e colocar o resultado em uma string s3, tal como o apresentado no **Exercício 12**, mas o i-ésimo elemento de s3 pode ser uma cópia de s1 ou s2, a depender de um sorteio aleatório em que existe 50% de chance de ser um elemento de s1 ou de s2. Assim, é necessário usar uma variável aleatória p tal que se seu valor é menor ou igual que 0.5, o caractere será de s1. Senão, será de s2. Ao terminar uma das strings, termine o programa. No exemplo do **Exercício 12**:

S1=	l	o	c	a	l			
	↔	↔	↔	↔	↔			

S2=	m	i	s	t	e	r	i	o
P=	0.18	0.50	0.70	0.59	0.93			
S3=	l	o	s	t	e			
	0	1	2	3	4			

Exercício 14: Um operador de **crossover** pode ser aplicado a duas strings s1 e s2 e consiste em se sortear aleatoriamente um ponto de s1 e s2. Escolhido este ponto, então, é realizada a troca de informações de s1 e s2 tal como mostrado no esquema da Figura 1.

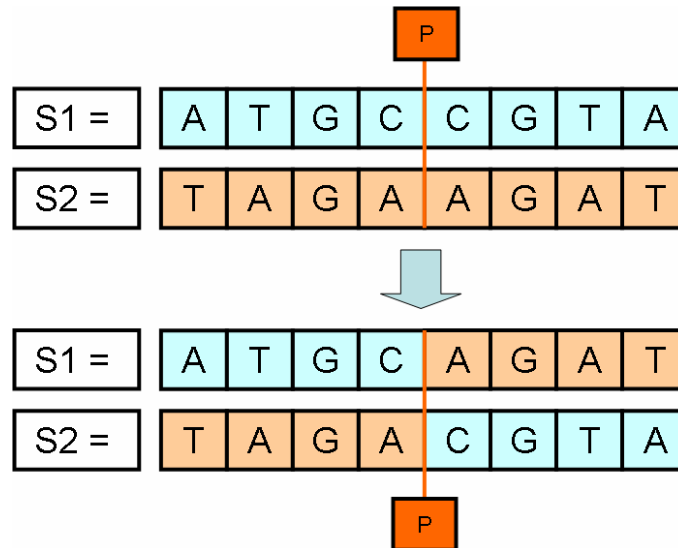


Figura 1: Aplicação do operador de **crossover** entre duas strings s1 e s2.

Construir um programa que:

- Realiza a leitura de duas strings s1 e s2.
- Emprega o operador de crossover para construir novas strings s1 e s2.
- Mostra as novas strings s1 e s2 e o valor do ponto p sorteado aleatoriamente que representa o índice a partir do qual ocorreu a troca de informações entre s1 e s2.

Exercício 15: Uma string é utilizada para representar uma das fitas de uma cadeia de DNA. Para tanto, as bases Adenina, Guanina, Citosina, Timina e Uracila são representadas pelas letras A, G, C, T e U, respectivamente. Deseja-se construir um programa que dada uma sequência de DNA é fornecida a sequência de RNA-m equivalente de acordo com a transformação indicada na Tabela 1.

TABELA 1

DNA	RNA-m
A	U
G	C
C	G
T	A

Teste o seu programa para a seguinte fita de uma cadeia de DNA:

A	T	C	C	G	T	T	A	A
---	---	---	---	---	---	---	---	---