

Programação I

Strings (ficha 12)

NOTA: Não utilizar funções que manipulam strings na resolução dos problemas. Os métodos `lower()` e `upper()` podem ser utilizados.

1. Implemente a função `int tamanho(char s[])` que devolve o tamanho da string `s`.
2. Implemente a função `void letras(char s[])` que recebe uma string como argumento e escreve uma letra por linha. Implemente uma versão com ciclo `while` e outra com ciclo `for`. Por exemplo, a chamada da função `letras("Aula")` deve mostrar:

```
A
u
l
a
```

3. Implemente a função `void letras_inv(char s[])` que recebe uma string como argumento e escreve as letras por ordem inversa (do fim para o início). Por exemplo, a chamada da função `letras_inv("Aula")` deve mostrar:

```
a
l
u
A
```

4. Implemente a função `void sub_cadeia(char s[])` que imprime a string numa escala invertida. Por exemplo, a chamada da função `sub_cadeia("Aula")` deve mostrar:

```
Aula
Aul
Au
A
```

5. Implemente a função `void espelho(char s[])` que recebe uma palavra e que a imprime seguida do seu reverso. Por exemplo, a chamada da função `espelho("Aula")` deve mostrar `AulaaluA`.
6. Um palíndromo é uma palavra que é escrita da mesma forma da esquerda para a direita ou da direita para a esquerda; por exemplo "aia" e "noon". Implemente a função `int palindromo(char s[])` que indica se a string `s` é um palíndromo ou não.
7. Implemente as seguintes funções sobre strings:
 - (a) `int vogal(char c)`: verifica se o argumento é uma vogal.
 - (b) `int conta_vogais(char s[])`: devolve o número de vogais existentes na string recebida como argumento. Utilize a função definida no ponto anterior.

8. Implemente a função `int ocorrencias(char letra, char string[])` que conta o n^o de vezes que `letra` aparece em `string`. Por exemplo, a chamada da função `ocorrencias("aula")` deve devolver 2.
9. Implemente a função `int ocorrencias2(char letra, char string[], int indice)` que conta o n^o de vezes que `letra` aparece na `string`. `indice` indica o índice a partir do qual a contagem deve ser feita. Por exemplo, a chamada da função `ocorrencias2('u', "aula", 2)` deve devolver 0.
10. Implemente a função `ocorrencias3(char letra, char string[], int indice)` que conta o n^o de vezes que `letra` aparece na `string` a partir de `indice`. A contagem não deve fazer distinção entre maiúsculas e minúsculas. Por exemplo, o resultado da chamada da função `ocorrencias3('a', "Aula", 0)` deve ser 2.
11. Crie uma função que tem como parâmetro uma frase (*string*) e conta o número de palavras dessa frase. Considere que as palavras estão separadas por espaços ou vírgulas. Por exemplo, `conta_palavras("computador, caderno e caneta")` devolve 4.
12. Uma empresa de eletrodomésticos pretende informatizar os dados relativos aos artigos que dispõe.
 - (a) Defina uma estrutura, `artigo`, adequada à representação de um artigo, contendo a seguinte informação: tipo do artigo, marca, modelo, preço e quantidade disponível no armazém
 - (b) Implemente uma função que lê um elemento do tipo artigo introduzido pelo utilizador: `struct artigo le_artigo()`.
 - (c) Considere que a informação sobre os artigos (em número menor que 1000) foi lida para um vetor de `artigo`. Implemente a função `total_artigo(struct artigo armazem[], int sz, char marca[], char modelo[])` que devolve o n^o total de artigos de uma determinada marca e modelo.
 - (d) Implemente o procedimento `void alerta_artigos(struct artigo armazem[], in sz)` que lista todos os produtos cuja existência em stock é inferior a 10 unidades.