FICHA DE TRABALHO			Nr. 5
Curso:	Técnico de Gestão e Programação de Sistemas Informáticos	**2022/2023**	
Disciplina:	Programação e Sistemas de Informação	**Turma:**	10ºE
Módulo:	4 – Estruturas de Dados Estáticas	**Data:**	Escolher.
Nome do aluno:		**Nr.:**	

Biblioteca <string>

MEDIÇÃO

- *length()* – número de caracteres na string;
- *clear()* – apaga o conteúdo de uma string;
- *empty()* – verifica se a string está vazia;

EXEMPLO

Copie e complete o seguinte programa, construindo a função **conteudo(string s).**

```
#include <iostream>
#include <string>
using namespace std;

void conteudo(string s)
{
// função que imprime no ecrã a mensagem
// "Com conteúdo (n caracteres)" ou "Sem conteúdo (0 caracteres)"
// conforme a string tenha ou não conteúdo
}

int main()
{
    string cumprimento;       // declaração
    cumprimento = "Olá!";     // atribuição
    conteudo(cumprimento);    // imprime "Com conteúdo (4 caracteres)"
    cumprimento.clear();      // limpeza
    conteudo(cumprimento);    // imprime "Sem conteúdo (0 caracteres)"
    cumprimento = "Bom dia!"; // Reatribuição
    conteudo(cumprimento);
}
```


Cofinanciado por: ANQEP SELO DE CONFORMIDADE EQAVET POCH PORTUGAL 2020 UNIÃO EUROPEIA

COMPARAÇÃO

- *compare()* – compara os conteúdos (totais ou parciais) de duas cadeias de caracteres alfanuméricos. Tem os seguintes argumentos:

Argumento	Descrição
pos	Posição inicial no processo de comparação. A indexação é iniciada em 0, pelo que o primeiro caractere é de índice 0, o segundo de índice 1, etc.
len	Número de caracteres que deverão ser comparados.
str	Texto de comparação.

Nota: caso o objetivo seja uma comparação completa entre duas cadeias de caracteres alfanuméricos, apenas é especificado o argumento *str*. Os argumentos *len* e *pos* são usados apenas em comparações parciais.

EXEMPLO

Copie, analise o código, execute e reflita sobre o resultado.

```
1  #include <iostream>
2  #include <string>
3  #include <locale>
4  using namespace std;
5
6  int main() {
7      setlocale(LC_ALL, "pt_PT.utf8");
8      string a1 = "gato preto";
9      string a2 = "gato branco";
10     if (a1.compare(a2) != 0) {
11         cout << "Os animais são diferentes\n";
12     }
13     if (a1.compare(0, 4, "gato") == 0) {
14         cout << "O 1º animal é um gato\n";
15     }
16     if (a1.compare(0, 3, "cão") != 0) {
17         cout << "O 1º animal não é um cão\n";
18     }
19     if(a1.compare(0, 4, a2, 0, 4) == 0) {
20         cout << "O 1º animal é um gato\n";
21     }
22 }
```

EXTRAÇÃO

- *substr()* – permite extrair um excerto de texto da cadeia de caracteres especificada. Deverão ser utilizados os argumentos seguintes:

Argumento	Descrição
pos	Posição inicial no processo de obtenção. A indexação é iniciada em 0. Todos os caracteres à esquerda da posição especificada não surgem na cadeia de caracteres alfanuméricos final.
len	Número de caracteres a ser devolvidos a partir da posição especificada.

Nota: caso o objetivo seja uma comparação completa entre duas cadeias de caracteres alfanuméricos, apenas é especificado o argumento *str*. Os argumentos *len* e *pos* são usados apenas em comparações parciais

EXEMPLO

Copie, analise o código, execute e reflita sobre o resultado.

```cpp
#include <iostream>
#include <string>
#include <locale>
using namespace std;

int main() {
    setlocale(LC_ALL, "pt_PT.utf8");
    string data;
    data = "15/03/2023";
    string dia, mes, ano;
    dia = data.substr(0, 2);
    mes = data.substr(3, 2);
    ano = data.substr(6, 4);
    // Impressão
    cout << "Data: " << data << endl; // Imprime "Data: 15/03/2023"
    cout << "Dia: " << dia << endl;   // Imprime "Dia: 15"
    cout << "Mês: " << mes << endl;   // Imprime "Mês: 03"
    cout << "Ano: " << ano << endl;   // Imprime "Ano: 2023"
}
```


Cofinanciado por: ANQEP SELO DE CONFORMIDADE EQAVET POCH PORTUGAL 2020 UNIÃO EUROPEIA

SUBSTITUIÇÃO

- *replace()* – permite substituir um excerto de texto por outro podendo também servir para acrescentar ou excluir conteúdo numa posição especificada. Deverão ser utilizados os argumentos seguintes:

Argumento	Descrição
pos	Posição a partir da qual se inicia a substituição. A indexação começa em 0.
len	Número de caracteres a serem substituídos.
str	Texto de substituição.

EXEMPLO

Copie, analise o código, execute e reflita sobre o resultado.

```
#include <iostream>
#include <string>
#include <locale>
using namespace std;

int main() {
    setlocale(LC_ALL, "pt_PT.utf8");
    string valor = "Eis um texto.";
    cout << valor << endl;                    // Imprime "Eis um texto."
    valor = valor.replace(7, 0, "pouco de "); // substitui "Eis um []texto."
                                              // por "Eis um [pouco de ]texto."
    cout << valor << endl;                    // Imprime "Eis um pouco de texto."

    valor = valor.replace(7, 8, "pequeno"); // substitui "Eis um [pouco de] texto."
                                            // por "Eis um [pequeno] texto."

    cout << valor << endl;                  // Imprime "Eis um pequeno texto."

    valor = valor.replace(7, 8, "");        // substitui "Eis um [pequeno] texto."
                                            // por "Eis um [] texto."

    cout << valor << endl;                  // Imprime "Eis um texto."
}
```

PESQUISA

- *find()* – permite encontrar um caractere ou um conjunto de caracteres numa cadeia de caracteres alfanuméricos especificada. Retorna um número inteiro (tipo int) que corresponde à posição indexada à primeira ocorrência. São usados os argumentos seguintes:

Argumento	Descrição
str	Cadeia de caracteres alfanuméricos a ser pesquisada
pos	Posição a partir da qual a pesquisa é iniciada. Caso não seja especificado este argumento, a pesquisa será iniciada a partir do primeiro caractere.

Nota: A função *find* opera em modo "*case-sensitive*" e retorna -1 caso não se verifique qualquer ocorrência.

EXEMPLO

Copie, analise o código, execute e reflita sobre o resultado.

```
1  #include <iostream>
2  #include <string>
3  #include <locale>
4  using namespace std;
5
6  int main() {
7      setlocale(LC_ALL, "pt_PT.utf8");
8
9      string valor = "Bom dia!";
10     cout << valor << endl;          // Imprime "Bom dia!"
11
12     int pesquisa;
13
14     pesquisa = valor.find("!");
15     cout << pesquisa << endl;       // imprime 7
16
17     pesquisa = valor.find("bom");
18     cout << pesquisa << endl;       // imprime -1
19
20     pesquisa = valor.find("dia");
21     cout << pesquisa << endl;       // imprime 4
22 }
```


Cofinanciado por: ANQEP SELO DE CONFORMIDADE EQAVET POCH PORTUGAL 2020 UNIÃO EUROPEIA

1. Considere a seguinte sequência de caracteres para os exercícios seguintes, que deve resolver todos no mesmo programa:

"era␣uma␣vez␣␣␣um␣gato␣maltês␣␣tocava␣piano␣e␣falava␣francês␣␣"

Nota: O caractere ␣ representa o espaço.

1.1. Inicialize a *string* ***texto***, declarada localmente na *main()*, com a sequência anterior.

1.2. Construa a função ***void mostra_palavras(string s)*** que adicione uma nova-linha ao ecrã e mostre a o conteúdo da string, uma palavra em cada linha do ecrã.

1.3. Construa a função ***int conta_chars(string s, char c)***, que determine o número de ocorrências de um caractere c na string s (ambos recebidos como argumentos).

1.4. Construa a função ***int substitui(string s, char c1, char c2)***, que substitua todas as ocorrências de ***c1*** por ***c2*** na string ***s*** (***s***, ***c1*** e ***c2*** todos recebidos como argumentos) e retorne quantas substituições foram efetuadas (Por exemplo, se a função devolver zero significa que não foi feita substituição alguma).

1.5. Construa a função ***void primeiras_maiusculas(string s)***, que substitua todas as ocorrência de cada primeira letra de uma palavra pela respetiva maiúscula.

1.6. Construa a função ***void troca_duplo_espaco(string s)***, que altere as sequências de dois espaços seguidos " " pela sequencia dos caracteres '.' e '\n' (ponto final e nova-linha).

1.7. Construa a função ***int conta_palavras(string s)***, que determine o número de palavras na string. Nota: Pode considerar uma palavra toda a sequência de caracteres limitado por espaços (com um espaço antes e um espaço depois).