

# Exercícios - Bibliotecas Padrão

---

O C como vimos em aula possui diversas bibliotecas disponíveis para uso, dentre elas `stdio.h`, `stdlib.h`, `string.h`, `ctype.h`, `math.h`, dentre outras. Os exercícios abaixo visam fixar os conceitos visto em sala.

## DICAS

---

Como vimos as bibliotecas são construídas em torno do conceito de função, sendo assim podemos a qualquer momento na internet verificar as funções disponíveis em uma biblioteca, como elas se comportam e como utilizá-las. Uma função pode ser invocada através seu nome seguido de 0 ou mais parametros entre parenteses e um ponto e vírgula. Um exemplo, se quisermos calcular o cosseno de um número, primeiro incluímos a biblioteca matemática nas primeiras linhas do nosso código com `#include <math.h>`, depois onde precisarmos, podemos chamar a função `cos()` e atribuir o valor a uma variável exemplo:

Na internet vc encontrará facilmente exemplos de como se usar as funções descritas neste exercício.

```
#include <stdio.h>
#include <math.h>

int main() {

    float resultado;
    resultado = cos(1);
}
```

Muitas outras funções estão disponíveis na biblioteca matemática e muito mais em outros bibliotecas.

Com base no exposto e no visto em sala de aula responda as questões a seguir.

## math.h

`math.h` contém as funções matemática, para os exercícios abaixo você precisará das seguintes funções:

```
sin()    -> Calcula o seno de um número, ex; x = sin(0.5);
```

```
cos()    -> Calcula o cos de um número, ex: x = cos(1);
log10()  -> Logaritmo na base 10, ex: x = log10(30);
log()    -> Logaritmo na base natural, ex: x = log(4);
sqrt()   -> Raiz quadrada de um número, ex: x = sqrt(64);
pow()    -> Potenciação, ex: x = pow(2, 3);
```

1. Escreva um programa imprima um menu na tela, com opções variando de 0 a 5, onde cada opção representa uma das operações mostrada acima. Logo em seguida o programa deverá ler do usuário uma opção e um valor. Aplique a função escolhida ao valor digitado e imprima na tela o nome da função que foi executada e resultado na tela.
2. Escreva um programa que possua o mesmo menu do programa excetuando a função, que leia a opção e leia dois inteiros que compõem um intervalo. Depois imprima uma tabela da função escolhida, onde cada linha da tabela é o resultado do valor aplicado à função acrescido de 0.5 a cada iteração do `for` até o final do intervalo. Exemplo, se o usuário escolher a opção correspondente ao `sen()`, e os números 1 e 3, o programa deverá imprimir:

```
0 seno de 1 é ...
0 seno de 1.5 é ...
0 seno de 2 é ...
0 seno de 2.5 é ...
0 seno de 3 é ...
```

onde a reticências é o resultado retornado pela chamada da função.

## string.h

A biblioteca `string.h` manipula strings. Vimos como ler strings utilizando `fgets`, use-o para fazer os exercícios abaixo. Vimos também que o `fgets` adiciona o enter na string, logo para evitarmos que isso aconteça use o seguinte para fazer a leitura:

```
char linha[50];
fgets(linha, 50, stdin);
linha[strlen(linha) - 1] = '\\0';
```

Para que o exemplo acima funcione você tem que incluir a biblioteca `string.h`, pois utilizamos `strlen` para saber o tamanho da string, assim sabemos onde ela termina no nosso vetor, como o último caractere é sempre o `\\0`, o `\\n` estará sempre imediatamente na posição anterior, por isso do `- 1`.

3. Escreva um programa que leia uma string e armazene em um vetor. Verifique o tamanho da string lida, ela não pode ser maior que 20, caso seja, imprima um erro `string inválida! Número de caracteres maior que 20`.
4. Escreva um programa que leia duas strings e armazene uma em cada variável, depois troque os valores das variáveis. DICA: Use um terceiro vetor para armazenamento temporário, exercício semelhante ao que fizemos para trocar números. Use `strcpy` para copiar o valor de uma string para a outra. `strcpy(string1, string2)` copia o valor da `string1` para a `string2`.
5. Escreva um programa que leia um string para o nome, e uma outra para o sobrenome, utilize a função `strcat` para unir as duas, imprima o resultado na tela.
6. Faça um menu, onde a pessoa deve responder a uma pergunta com "sim" ou "não", caso seja não imprima na tela, "Ok, não respondido", caso seja sim reponda "Ok, sim!". Utilize a função `strcmp`.

## ctype.h

Ao incluir esta biblioteca você poderá usar uma série de funções que lidam com caracteres.

7. Faça um programa que leia uma string e verifique se todos os caracteres são compostos por dígitos decimais. Para isso utilize a função `isdigit()`, ela recebe como parâmetro um caracter, e retorna 0 se o caracter não é um dígito ou outro valor caso contrário. ex: `isdigit('c')`, resultado será 0, qualquer valor diferente de zero indica que é um dígito.
8. Faça um programa que leia uma string e faça com que todas as letras virem maiúsculas, para isso utilize a função `toupper()` para obter a letra em maiúscula, você deverá então substituir a letra obtida do vetor com a letra retornada pela função `toupper()`. ex. `toupper('c')`, irá retornar o valor equivalente para 'C', coloque então esse valor na posição atual do vetor que está percorrendo.
9. Pesquise mais sobre as funções em `ctype.h`, escreva um programa qualquer que use uma das funções não vistas aqui.