

# Tutorial – Linguagem C

Lucas Gasparello

May 28, 2017

## 1 Introdução

Esse guia é baseado no livro C How to Program, H. M. Deitel, foi escrito sem nenhum rigor, ele é puramente destinado ao auxílio dos participantes do minicurso de Introdução a Arduino. Para um estudo mais detalhado da linguagem é recomendado a leitura desse livro. Nesse tutorial será apresentado noções básicas de variáveis, funções e estruturas de controle e repetição.

## 2 Variáveis

Programas servem, essencialmente, para manipular dados: ao executarmos um programa, fornecemos dados e esperamos que ele faça alguma coisa com eles. Os dados podem se apresentar em duas formas distintas: como constantes ou variáveis.

A declaração de uma variável possui a seguinte sintaxe:

```
tipo_da_variavel nome_da_variavel;
```

Os tipos de variáveis variam de acordo com o seu tamanho em bytes, podem ser inteiras, em ponto flutuante ou caracteres.

Além disso, C oferece operadores para as quatro operações aritméticas e também um operador para calcular o resto da divisão entre dois números inteiros, além do operador de igualdade, representados por  $+$ ,  $-$ ,  $/$ ,  $*$ ,  $\%$ ,  $=$ , respectivamente.

### LINGUAGEM C TIPOS DE DADOS

Tipo	Extensão de bit	Escala
char	8	0 a 255
int	16	-32768 a 32767
float	32	3.4e-38 a 3.4e+38
double	64	1.7e-308 a 1.7e+308
void	0	sem valor

TAMANHO E ESCALA DOS TIPOS DE DADOS

Figure 1: Tipos de variável

### 3 Funções

Uma função é um conjunto de operações para executar uma dada tarefa. A estrutura (sintaxe) de uma função é:

```
tipo_da_funcao nome_da_funcao(parametros)
{
    corpo_da_funcao;
}
```

O tipo da função remete aos tipos de variáveis apresentado anteriormente, caso a função não seja do tipo VOID, deve retornar algo, como no exemplo abaixo:

```
int soma(int a, int b){
    return (a + b);
}
```

Nesse exemplo a função recebe 2 paramentos e retorna a soma deles.

### 4 Estruturas de controle e repetição

Normalmente um programa em c é executado sequencialmente, ou seja, uma instrução após a outra. Para o programa decidir qual linha deve executar, ou seja, tomar decisões, são usadas estruturas de controle, como o if/else.

Se a condição for verdadeira o programa executa a instrução do if, caso seja falsa, executa o else. Basicamente o programa faz um escolha entre duas condições, como no exemplo abaixo.

```
if (nota >= 60){
    printf("passou");
}
else{
    printf("falhou");
}
```

Já para fazer o programa rodar um trecho de código até que uma condição seja falsa é usado os comandos While e For.

A sintaxe do while é:

```
while( condição)
{
    corpo_da_função;
}
```

Enquanto a condição for verdadeira o programa ficará "preso" nesse loop.

Abaixo é mostrado um exemplo que executa 10 vezes e soma o valor de i no total, ao final o valor de total será 55( 1+2+3+4...10).

```
while (i<=10){
    total = total + i;
    i++;
}
```

A sintaxe do for é:

```
for(valor inicial; condição, alteração){
    corpo_da_função;
}
```

Diferentemente do while, a estrutura for inicia e incrementa/decrementa uma variável na mesma linha, porém segue o mesmo princípio do while, enquanto a condição for verdadeira o programa fica "preso" no loop.

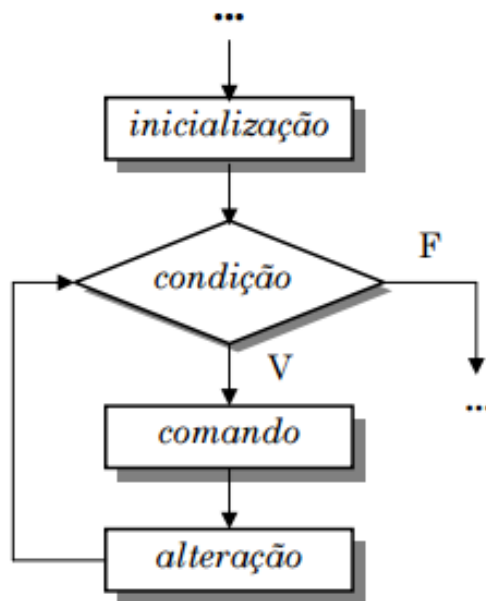


Figure 2: Estrutura For

```

for(i=1;i<=10;++i){
printf("%d", i);
}

```

O exemplo acima é um programa que imprime os números de 1 a 10 sequencialmente.

## 5 Desafios

- Faça um programa que identifique se um numero é primo.
- Faça um conversor de segundos para anos,dias,horas e segundos.

## References

- [1] DEITEL,Paul; DEITEL Harvey.**C: How to Program**. 7th Edition,Prentice Hall.