



Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica  
e de Computação



---

# EA871 - Laboratório de Programação Básica de Sistemas Digitais

Roteiro 1 - Apresentação da disciplina e Introdução à Programação em C

---

Aluno: Fernando Teodoro de Cillo

RA: 197029

Campinas  
Março de 2022

# 1 Introdução

A disciplina EA871 é voltada para a programação de microcontroladores, que é geralmente feita em C. A linguagem C é mais interessante para esse tipo de aplicação do que a linguagem Python aprendida previamente, e este primeiro roteiro serve como forma de aprender a programar em C tendo como base o que já foi ensinado sobre Python.

As principais vantagens de C, que é uma linguagem compilada, em relação a Python, que é interpretada, são: tempo de execução de programa menor; economia de memória no processador do microcontrolador por gerar programas executáveis e dispensar um interpretador; e melhor controle do programador sobre a alocação de memória.

## 2 Experimento

O experimento consiste em analisar as diferenças entre o Python e o C, através da comparação entre códigos semelhantes nas duas linguagens e da elaboração de novos códigos.

### 2.1 Variáveis de entrada e saída

A tabela 1 mostra as entradas e saídas de diversos códigos em Python e em C. (*n*) denota a solução *n*.

Tabela 1: Comparação entre entradas e saídas de programas em Python e C

	Python		C		
	Entrada	Saída	Entrada	Saída	&
Exercício 8	(1) n	(1) n; n_fat	(1) n	(1) n; fatorial	(1) n
	(2) n	(2) n; n_fat	(2) n	(2) n; nfat	(2) n
	(2) n	(2) n; n_fat	(3) n	(3) n; nfat	(3) n
Exercício 9	(1) n; i; j	(1) mult	(1) n; i; j	(1) n; i; j; mult	(1) n; i; j
	(2) n; i; j	(2) mult_i; mult_j	(2) n; i; j	(2) n; i; j; multi; multj	(2) n; i; j
	(2) n; i; j	(2) mult_i; mult_j	(3) n; i; j	(3) n; i; j; multi; multj	(3) n; i; j
	(2) n; i; j	(2) mult_i; mult_j	(4) n; i; j	(4) n; i; j; multi; multj	(4) n; i; j
Exercício 10	n	n; i	n	n; i	n
Exercício 11	n	n	n	n	n
pertence.*	n; x; y	x; y	n; x; y	x; y	n
ocor_palavras.*	frase; palavra	frase; palavra; m; n; numOcor	frase; palavra	frase; palavra; m; n; numOcor	-

### 2.2 Erros de compilação do arquivo *ex6.c*

A execução do código C *ex6.c* gera os erros da figura 1. Podemos notar que o erro está na linha 25, mostrada na figura 2.

```
Compilation failed due to following error(s).
main.c:25:17: error: stray '\302' in program
25 |     scanf("%d", a);
   |               ^
main.c:25:18: error: stray '\254' in program
25 |     scanf("%d", a);
   |               ^
main.c:25:19: error: 'a' undeclared (first use in this function)
25 |     scanf("%d", a);
   |               ^
main.c:25:19: note: each undeclared identifier is reported only once for each function it appears in
```

Figura 1: Captura de tela dos erros ao executar o programa

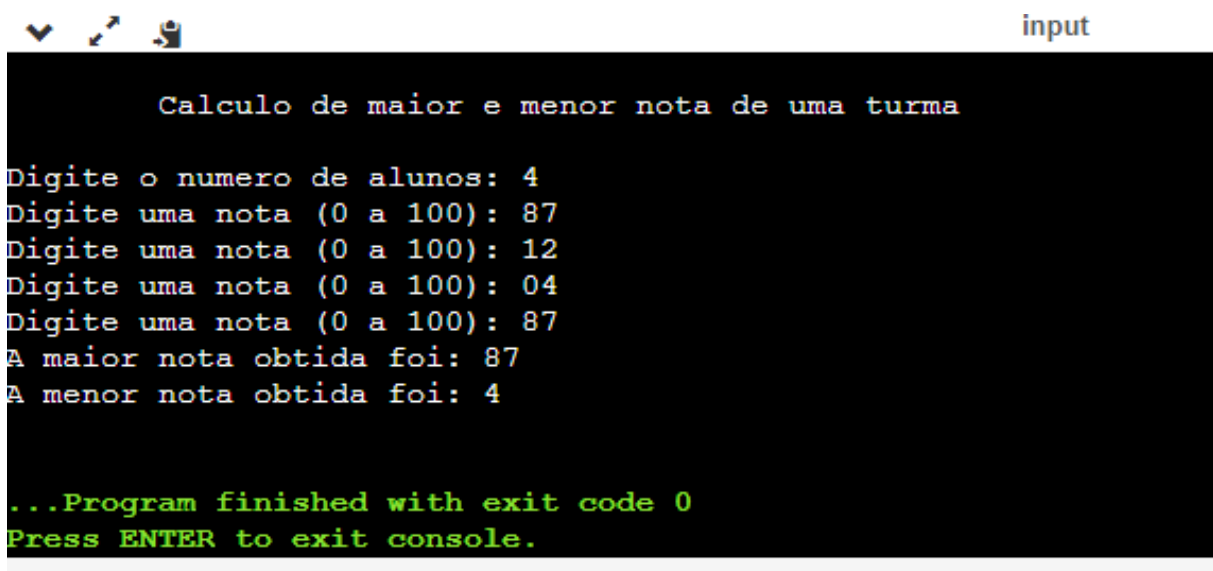
```

23     while (contador < n) {
24         printf("Digite uma nota (0 a 100): ");
25         scanf("%d", -a);
26         contador = contador + 1;
27         if (notamaior < nota)
28             notamaior = nota;
29         if (notamenor > nota)
30             notamenor = nota;
31     }

```

Figura 2: Bloco que contém o erro do código *ex6.c*

Os erros que ocorreram são devidos à sintaxe do segundo argumento da função `scanf()`. O operador `-` não é o utilizado para indicar um endereço, que era o esperado, e a variável `"a"` não foi declarada e nem é usada em nenhum momento no código, porque a variável utilizada para as notas é `"nota"`. Assim, substituindo `"-a"` por `"&nota"` na linha 25 o código executa sem erros. A figura 3 mostra o terminal do compilador online com as correções feitas.



```

input

Calculo de maior e menor nota de uma turma

Digite o numero de alunos: 4
Digite uma nota (0 a 100): 87
Digite uma nota (0 a 100): 12
Digite uma nota (0 a 100): 04
Digite uma nota (0 a 100): 87
A maior nota obtida foi: 87
A menor nota obtida foi: 4

...Program finished with exit code 0
Press ENTER to exit console.

```

Figura 3: Captura de tela do terminal com o código corrigido

## 2.3 Comparação entre Python e C

Comparando os arquivos *ocor\_palavras.py* e *ocor\_palavras.c* encontramos algumas diferenças fundamentais entre as duas linguagens, apesar de a função dos códigos ser exatamente a mesma (contar o número de ocorrências de uma determinada palavra em um texto).

Se na introdução deste roteiro foram apresentadas algumas vantagens da linguagem C em relação ao Python, aqui vemos algumas das desvantagens. O arquivo `.c` contém muito mais linhas de códigos que o arquivo `.py` porque a linguagem C exige que sejam declaradas as variáveis, além de necessitar utilizar diretivas para incluir outros arquivos no momento da compilação (operador `#`). Outra exigência de códigos em C é o uso de `;` ao fim de linhas, que não é necessário em Python, apesar de possível. Além disso, muitas funções imbutidas em Python não o são em C (o que economiza espaço de armazenamento do arquivo quando essas funções não são utilizadas, mais exige que elas sejam definidas pelo programador quando o forem).

A figura 4 mostra justamente essa programação de uma "função" (não foi definida como função no arquivo) que equivale à linha "numOcor = frase.count (palavra)" em Python. É notável que o método .count() do Python substitui 2 funções for, 3 funções if e uma função break em C, e executa igualmente porque implicitamente realiza as mesmas operações.

```

35 // conte a quantidade de ocorrências
36 for (i=0; i <= m-n; i++) {
37     if (frase[i] == palavra[0]) {
38         ocorre = 1;
39         for (j=1; j<n; j++) {
40             if (frase[i+j] != palavra[j]) {
41                 ocorre = 0;
42                 j++;
43                 break;
44             }
45         }
46         numOcor = numOcor+ocorre;
47         if (ocorre) i=i+j-1;
48     }
49 }

```

Figura 4: Bloco de código em C equivalente à linha "numOcor = frase.count (palavra)" em Python

## 2.4 Algoritmos

### 2.4.1 palindromo.c

Palíndromo é uma palavra que pode ser lida igualmente da direita para a esquerda ou da esquerda para a direita. Uma das maneiras de escrever um código que identifica um palíndromo pode ser vista na figura 5, em que a letra mais à esquerda que ainda não foi analisada é comparada com a letra mais à direita ainda não analisada. Somente metade do comprimento da palavra é necessária para a execução correta do laço for, já que duas letras são contabilizadas por iteração. As figuras 6a e 6b mostram a execução do código quando a palavra não é ou é um palíndromo, respectivamente.

```

1  /* EA871 - Laboratório de Programação Básica de Sistemas Digitais */
2  /* Fernando Teodoro de Cillo RA 197029 Turma Q */
3  /* Roteiro 1 */
4
5  #include <stdio.h>
6  #include <string.h> /* Biblioteca que permite realizar operações com strings */
7
8  int main()
9  {
10     char palavra[50]; /* A maior palavra registrada da língua portuguesa possui 46 letras => 50 caracteres bastam */
11
12     printf("VERIFICADOR DE PALINDROMOS: \n\n");
13     printf("Qual a palavra a ser analisada?\n\n");
14     scanf("%s", palavra); /* Possível palíndromo a ser verificado */
15     int carac = strlen(palavra); /* Número de caracteres na string é o tamanho da palavra */
16     printf("\n");
17
18     for (int cont = 0; cont < carac/2; cont++) { /* Como duas letras serão analisadas por iterações, só precisamos de metade */
19         if (palavra[cont] != palavra[carac - cont - 1]) { /* Verifica se a letra mais a esquerda difere da mais
20             a direita (desconsiderando as já verificadas) */
21             printf("%s não é palindromo", palavra); /* Se alguma diferença for encontrada, não é um palíndromo */
22             return 0;
23         }
24     }
25     printf("%s eh palindromo", palavra); /* Se não houver diferença, a palavra é igual de trás-para-frente (palíndromo) */
26     return 0;
27 }

```

Figura 5: palindromo.c

```

VERIFICADOR DE PALINDROMOS:

Qual a palavra a ser analisada?

fernando

fernando não é palindromo

...Program finished with exit code 0
Press ENTER to exit console.

```

(a) Palavra que não é palíndromo

```

VERIFICADOR DE PALINDROMOS:

Qual a palavra a ser analisada?

tenet

tenet eh palindromo

...Program finished with exit code 0
Press ENTER to exit console.

```

(b) Palavra que é palíndromo

Figura 6: Execução do código palindromo.c

#### 2.4.2 fahrenheit.c

A conversão entre diversas unidades para uma mesma grandeza é necessária para realizar operações em um determinado sistema (Sistema Internacional ou Sistema Imperial, por exemplo) de maneira que não ocorram erros de cálculo. Para encontrar um valor em graus Celsius equivalente a um valor em graus Fahrenheit é necessário realizar apenas operações de subtração (ou soma, na operação inversa), multiplicação e divisão. Um exemplo de código em C que realiza essa conversão é dado na figura 7, e sua execução na figura 8.

```

1  /* EA871 - Laboratório de Programação Básica de Sistemas Digitais */
2  /* Fernando Teodoro de Cillo    RA 197029    Turma Q */
3  /* Roteiro 1 */
4
5  #include <stdio.h>
6
7  int main() {
8      float celsius, fahrenheit; /* Definição da tipagem das variáveis */
9
10     printf("CONVERSOR DE TEMPERATURA: \n\n");
11     printf("Insira uma temperatura em Farenheit:\n\n");
12     scanf("%f", &fahrenheit); /* Armazena um valor (em °F) no endereço
13     reservado para a variável fahrenheit */
14
15     celsius = (fahrenheit-32) * 5/9; /* Converte para °C */
16
17     printf("%.2f F = %.2f C", fahrenheit, celsius);
18     /* %.2f identifica uma variável float com duas casas decimais */
19
20     return 0;
21 }

```

Figura 7: fahrenheit.c

```

CONVERSOR DE TEMPERATURA:

Insira uma temperatura em Farenheit:

-40
-40.00 F = -40.00 C

...Program finished with exit code 0
Press ENTER to exit console.

```

Figura 8: Execução do código fahrenheit.c