

Introdução à programação C/C++

aula prática 1¹

1 Exercícios

1.1 Exercício 1

Escreva um programa que defina uma variável do tipo inteiro, atribua à ela um valor qualquer e imprima este valor em decimal, octal e hexadecimal.

1.2 Exercício 2

Escreva um programa que defina variáveis dos tipos float, double e long double, atribua valores a estas variáveis e as imprima usando o formato normal e o de notação científica.

1.3 Exercício 3

Um número amigo n na faixa de 100 até 999 é um número que obedece a seguinte regra.

$$n = \text{centenas}^3 + \text{dezenas}^3 + \text{unidades}^3$$

Por exemplo, o número 153 é obedece a regra pois

$$153 = 1^3 + 5^3 + 3^3$$

Escreva um programa que leia uma variável inteira n ($100 \leq n \leq 999$). Considere que o valor lido está neste intervalo. Decomponha o valor lido em centenas, dezenas e unidades e imprima estes números e o resultado da fórmula acima.

Dica importante: Em C o operador % fornece o resto da divisão de dois operandos inteiros.

Considere, por exemplo, as variáveis inteiras a , b , c . A expressão $c = a \% b$; armazena em c o resto da divisão de a por b .

Exemplo de saída:

```
Número: 153
Centenas : 1
Dezenas : 5
Unidades : 3
Amigo : 153
```

¹ Este material é praticamente uma reprodução do produzido pelo Prof. Adriano Joaquim de Oliveira Cruz, disponível em <http://equipe.nce.ufrj.br/adriano/c/praticas.htm>

1.4 Exercício 4

É possível em C usar caracteres em expressões aritméticas. Por exemplo, é possível escrever o código mostrado abaixo.

```
char car = 'a';
car = car + 1;
printf (" %c \n ", car); /* imprime o caractere seguinte ao 'a' */
```

Escreva um programa que defina uma variável do tipo `char`, atribua a esta variável um caractere qualquer e imprima o caractere seguinte e o anterior daquele armazenado em `c`. Verifique qual é o caractere anterior ao `'a'` e qual é o seguinte ao `'z'`.

1.5 Exercício 5

Considere que em um programa um ângulo como `450 : 25' : 30"` é armazenado em três variáveis inteiras diferentes, uma para os graus, outra para minutos e outra para segundos. Escreva um programa que leia dois ângulos e imprima a sua soma.

```
Primeiro ângulo:
graus?
45
minutos?
25
segundos?
30
Segundo ângulo:
graus?
1
minutos?
50
segundos:
30
Soma:
47 16 0
```

1.6 Exercício 6

Escreva um programa que leia um valor inteiro positivo (*colunas*) do teclado e imprima uma linha com *colunas* asteriscos. Não é necessário testar se o número é positivo.

```
Exemplo de funcionamento:
ze@recreio~$ ./ex1
Entre com um numero positivo
7
*****
ze@recreio~$
```

1.7 Exercício 7

Modifique o programa anterior de modo que se o usuário digitar um número negativo o programa imprima um aviso e termine.

Exemplo de funcionamento:

```
ze@recreio~$ ./ex2
Entre com um numero positivo
-7
Valor invalido.
ze@recreio~$
```

1.8 Exercício 8

Modifique o programa anterior de modo que caso o usuário digite um número negativo o programa entre um laço de repetição até um número positivo ser digitado.

Exemplo de funcionamento:

```
ze@recreio~$ ./ex3
Entre com um numero positivo
-7
Valor invalido.
Entre com um numero positivo
-2
Valor invalido.
Entre com um numero positivo
13
*****
ze@recreio~$
```