

# Introdução à programação C/C++

## aula prática 4<sup>1</sup>

### 1 Exercício 1

Complete o programa mostrado na listagem 1. Este programa lê uma série de números inteiros e calcula e imprime os fatoriais. A série termina quando um número negativo é lido.

Listagem 1

```
#include <stdio.h>

long int fat ( long int n );

int main ( void ) {
    long int numero;

    while(1){
        scanf("%ld", &numero);
        if (numero < 0){
            break;
        }
        printf("O fatorial de %ld vale %ld\n", numero, fat(numero));
    }
    return 0;
}

long int fat(long int n){
}
```

### 2 Exercício 2

Complete o programa mostrado na listagem 2. Este programa calcula a combinação de  $n$  elementos tomados  $p$  a  $p$ , que é dada pela fórmula

$$C_n^p = \frac{n!}{(n-p)! \times p!} \quad (1)$$

Por exemplo, com esta fórmula podemos calcular de quantas maneiras diferentes podemos juntar 10 cartas de um baralho com 40, ou  $C_{40}^{10}$ .

---

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>

```
#include<stdio.h>
long int fat(long int);
long int c (long int n, long int p);

int main(void){
    long int n, p;
    while(1){
        scanf("%ld %ld", &n, &p);
        if (n < p ){
            break;
        }
        printf("O valor da combinação de %ld %ld a %ld vale %ld",
               n, p, p, c(n, p));
    }
    return 0;
}

long int fat(long int n){
}

long int c (long int n, long int p){
}
```

### 3 Exercício 3

Considera a função:

```
int quantasVezes(char frase[], char procurado);
```

Essa função retorna quantas vezes o caractere (**procurado**) apareceu em uma cadeia de caracteres **frase**. O trecho de programa abaixo imprime quantas vezes apareceu em uma frase um determinado caractere.

```
char c;
char frase[80];
int vezes;

/* ... */

gets(frase);
c = 'a';
vezes = quantasVezes(frase, c);
printf("O caracter %c apareceu %d vezes \n", c, vezes);

/* ... */
```

Escreva a função **quantasVezes** e um programa que, usando esta função, imprima a frequência das vogais em uma frase.

## 4 Exercício 4

Escreva um programa que leia uma frase de até 80 caracteres e a imprima após alterá-la segundo a criptografia de César. Neste método, cada letra da frase passa a ser a letra seguinte do alfabeto. A letra 'Z' via letra 'A'.

**Importante:** Somente letras podem ser modificadas. Não utilize frases com acentos. Letras maiúsculas continuam maiúsculas e o mesmo acontece com letras minúsculas.

Neste exercício, você deve criar a função `void criptografa(char frase[])`.

## 5 Exercício 5

Escreva uma função `void decriptografa(char frase[])` que recebe uma frase criptografada segundo o algoritmo mostrado no exercício anterior e decriptografa. Escreva um programa que usa esta função.

## 6 Exercício 6

Considere que você está controlando um robô que se move em um mundo plano de largura e altura definidas (Figura 1). A sua tarefa é completar o programa mostrado na listagem 3. Este programa lê os comandos do teclado e altera as posições do robô. Observe que se você emitir uma ordem inválida o robô não se move. Por exemplo, veja na listagem que se ele está no limite direito do mundo e você deu uma ordem para ele andar para a direita, neste caso o robô fica parado.

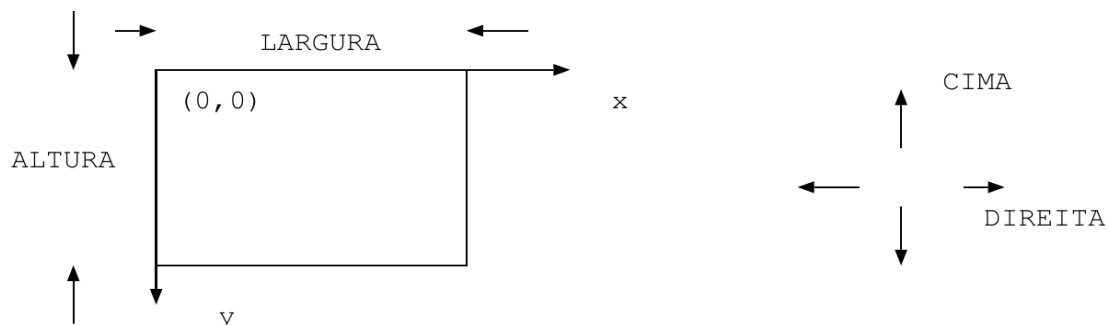


Figura 1: Mundo Plano

Listagem 3

```
#include<stdio.h>
#define ALTURA 10
#define LARGURA 10

int moveEsquerda(int x);
int moveDireita(int x);
int moveCima(int x);
int moveBaixo(int x);
```

```

int main (void){
    int px = 0 , py = 0;
    char ordem;
    int continua = 1;

    while (continua){
        __fpurge(stdin); //no windows: fflush(stdin)
        ordem = getchar();
        switch (ordem) {
            case 'a': case 'A':
                px = moveEsquerda(px);
                break;
            case 'd': case 'D':
                px = moveDireita(px);
                break;
            case 'w': case 'W':
                py = moveCima(py);
                break;
            case 'x': case 'X':
                py = moveBaixo(py);
                break;
            case 'q' : case 'Q':
                continua = 0;
                break;
            default:
                puts("Opção inválida");
        }
        printf("Estou em %d %d\n", px, py);
    }
    return 0;
}

int moveEsquerda(int x){}
int moveDireita(int x){
    if (x < LARGURA){
        x++;
    }
    return x;
}

int moveCima(int y){}
int moveBaixo(int y){}

```

## 7 Exercício 7

Uma das maneiras de calcular o valor de  $\pi$  é usar a série

$$\frac{\pi}{4} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} \dots \quad (2)$$

Escreva um programa que leia o número de termos da série que o usuário deseja usar para calcular  $\pi$  e imprima o valor calculado. Para calcular o valor defina uma função que tenha o seguinte protótipo:

```
double pi(int termos);
```