

# Prova-02

Prof. Msc. Elias Batista Ferreira  
Prof. Dr. Gustavo Teodoro Laureano  
Profa. Dra. Luciana Berretta  
Prof. Dr. Thierson Rosa Couto

## Sumário

<b>1</b>	<b>Número de dígitos (++)</b>	<b>2</b>
<b>2</b>	<b>Triângulo de Pascal (++)</b>	<b>3</b>
<b>3</b>	<b>Potência prima (+++)</b>	<b>4</b>

# 1 Número de dígitos (++)



(++)

Escreva um programa que leia um número  $n$  e apresente a quantidade de dígitos que ele possui.

Você deve implementar a função:

```
1 /**
2  * Funcao que calcula a quantidade de digitos de um numero inteiro
3  * @param n um numero inteiro
4  * @return quantidade de digitos de n
5  */
6 int digit_count(long int n);
```

## Entrada

O programa deve ler um número inteiro  $n$ .

## Saída

O programa deve apresentar uma linha com a mensagem: "Numero de digitos:  $c$ ", onde  $c$  é a quantidade de dígitos de  $n$ .

## Exemplo

Entrada	Saída
123	Numero de digitos: 3

## 2 Triângulo de Pascal (++)



(++)

Faça um programa que calcule e apresente uma faixa de linhas do Triângulo de Pascal. Cada linha do Triângulo de Pascal é dado pela seguinte equação:

$$\text{Linha } n: \binom{n}{0} \quad \binom{n}{1} \quad \binom{n}{2} \quad \dots \quad \binom{n}{n} \quad (1)$$

sendo que

$$\binom{n}{p} = C_{n,p} = \frac{n!}{p!(n-p)!} \quad (2)$$

### Entrada

O programa deverá ler 2 números inteiros, sendo o primeiro correspondendo à linha inicial e o segundo à linha final do Triângulo de Pascal.

### Saída

O programa deverá imprimir as linhas do Triângulo de Pascal com os números separados por vírgula.

### Exemplo

Entrada	Saída
3 4	1, 3, 3, 1 1, 4, 6, 4, 1

  

Entrada	Saída
0 5	1 1, 1 1, 2, 1 1, 3, 3, 1 1, 4, 6, 4, 1 1, 5, 10, 10, 5, 1

### 3 Potência prima (+++)



(+++)

Uma potência prima  $n$  é um número maior que 1 que é escrito por uma potência de um único número primo. Ou seja,  $n = k^p$ , sendo  $k$  um número primo e maior do que 1. Exemplos de potências primas:  $7 = 7^1$ ,  $4 = 2^2$ ,  $25 = 5^2$ ,  $27 = 3^3$  e  $81 = 3^4$ . Escreva um programa que imprima os  $N$  primeiros termos da sequência de potências primas e suas representações em potências primas correspondentes.

Você deve implementar a função:

```
1  /**
2   * Função que verifica e decompõe um número n como uma potência prima. Esta
3   * função recebe o valor n e retorna o resultado da decomposição de n como uma
4   * potência prima via ponteiros. Quando a decomposição é possível, a função
5   * retorna 1.
6   * Quando a decomposição não é possível, a função retorna 0.
7   *
8   * @param n valor inteiro a ser verificado
9   * @param k ponteiro para a base da potência prima.
10  * @param p valor da potencia.
11  * @return retorna 1 caso n seja uma potência prima e 0 caso contrário.
12  * no último caso, quando n não é uma potência prima, os conteúdos de k
13  * e p devem ser desconsiderados.
14  */
15  int potencia_prima( int n, int * k, int * p );
```

#### Entrada

O programa deve ler um número inteiro  $N$  referente a quantidade de termos da sequência de potências primas.

#### Saída

O programa deve apresentar  $N$  linhas, cada uma correspondendo a uma potência prima e sua representação correspondente. A saída deve seguir o formato  $n : k^p$ .

#### Dica

Para facilitar, você pode construir outras funções auxiliares. Quando for imprimir o caracter  $^$ , certifique-se que ele foi digitado pelo teclado, ou seja, não coloque o símbolo via Ctrl+C | Ctrl+V.

#### Exemplo

Entrada	Saída
7	2 : 2^1 3 : 3^1 4 : 2^2 5 : 5^1 7 : 7^1 8 : 2^3 9 : 3^2