Linguagens de Programação 1

Francisco Sant'Anna
Sala 6020-B
francisco@ime.uerj.br

http://github.com/fsantanna-uerj/LP1

Funções

```python
def celsius_para_fahrenheit (c):
    f = c * 9/5 + 32
    return f
```

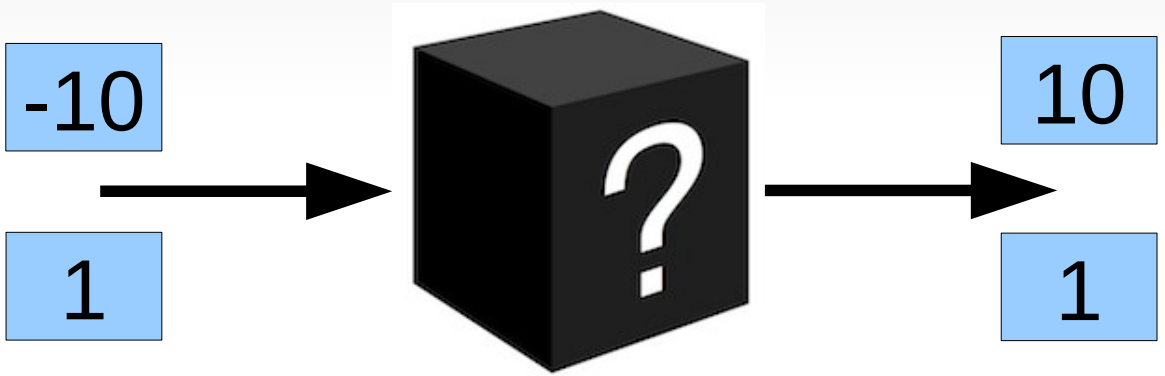
```python
C = input()
F = celsius_para_fahrenheit(C)
print(F)
```

```c
int celsius_para_fahrenheit (int c) {
    int f = c * 9/5 + 32;
    return f;
}
```

```c
int C;
scanf("%d", &C);
int F = celsius_para_fahrenheit(C);
printf("%d\n", F);
```

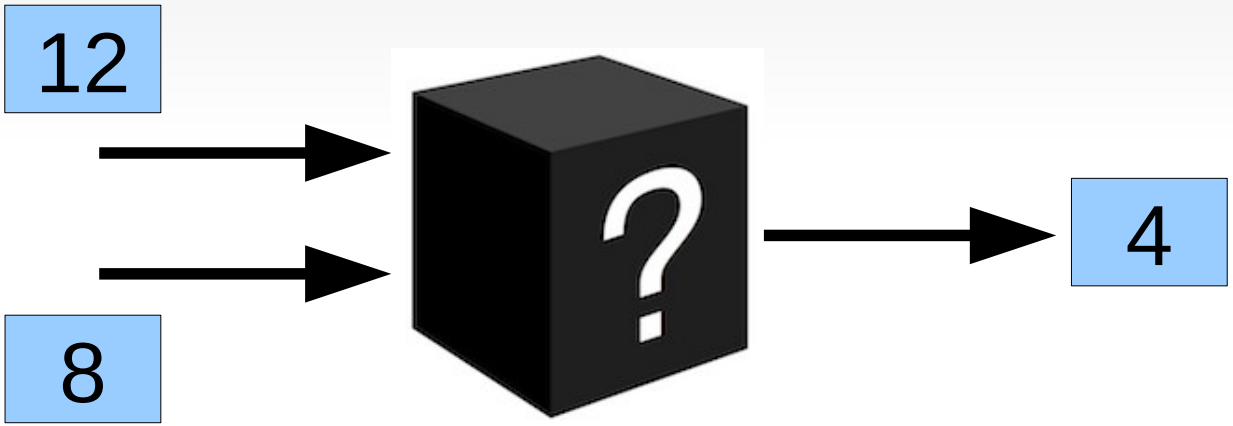
função **abs**:

- recebe um inteiro
- retorna o seu valor absoluto



função **mdc**:

- recebe dois inteiros
- retorna o *mdc* entre eles



Funções

- São abstrações de código
- Uma função representa uma computação complexa através de um nome
- Vantagens
  - Tornam o código mais legível e organizado
  - Permitem o reuso de código
  - Tornam a localização e correção de erros mais fácil

Exercício 4.1

- Crie uma função `eh_primo` que recebe um inteiro `n` e retorna se ele é primo ou não.
  - `eh_primo(4) -> 0`
  - `eh_primo(7) -> 1`
- Dicas:
  - A operação `%` calcula o resto entre dois valores:
    - `11 % 4 = 3`
  - Em C, tipicamente usamos `1` e `0` para representar *verdadeiro* e *falso*, respectivamente.

Exercício 4.2

- Crie uma função `todos_os_primos` que recebe um inteiro `max` e exibe todos os primos entre `1` e `max`.
- Usar a função do exercício 4.1 sem alterá-la.

TODO: protótipos e modulariazação