

Programação Estruturada

Introdução ao curso

Professores Emílio Francesquini e Carla Negri Lintzmayer

2018.Q3

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC



O que vamos aprender neste curso

O que vamos aprender neste curso

- Como usar um computador para resolver problemas
 1. Definiremos um problema a ser resolvido
 2. Discutiremos uma solução para o problema
 3. Descreveremos um **algoritmo** para resolver o problema
 - Sequência **bem definida** de comandos e passos
 4. **Implementaremos** este algoritmo e criaremos um **programa**
 - Sequência de comandos e passos que um computador deve executar

Problema: Dado um número x , quais são os divisores de x ?

Discussão: Os candidatos a serem divisores de x só podem estar no conjunto $\{1, 2, 3, \dots, x\}$. Podemos testar cada um desses números e verificar se eles dividem x .

Algoritmo:

1. Para todos os números inteiros $r \in \{1, 2, \dots, x\}$:
 - 1.1 Se o resto da divisão de x por r é zero, então r é divisor de x

O que vamos aprender neste curso

Problema: Dado um número x , quais são os divisores de x ?

Implementação:

```
1  #include <stdio.h>
2
3  int main() {
4      int x, i;
5      scanf("%d", &x); // lendo um numero do teclado
6      for (i = 1; i <= x; i++) {
7          if (x % i == 0)
8              printf("%d\n", i); // imprimindo um divisor
9      }
10     return 0;
11 }
```

O que vamos aprender neste curso

- Um algoritmo pode ser descrito de várias formas, como em português ou em uma **linguagem de programação**
 - Conjunto de instruções e regras para gerar um programa de computador
 - Usadas para criar arquivos de texto comum: códigos-fonte
- Aprenderemos a **implementar** um algoritmo em linguagem **C**
- A vantagem de se implementar um algoritmo em uma linguagem de programação é que podemos, a partir daí, criar um **programa** que usa o computador para resolver o problema

Por que aprender algoritmos e programação?

- Atividade básica de qualquer bom computeiro
- Para ser capaz de automatizar algum processo
- Para criar ferramentas/protótipos, você deverá fazer simulações para a realização de testes preliminares
- Para enxergar situações onde uma solução computacional pode trazer benefícios
- Para testar hipóteses
- Para resolver sistemas complexos de equações que não necessariamente podem ser resolvidos por softwares padrões (como MatLab)
- Posso ter algum retorno financeiro com isso!
- Porque é legal e desafiador!

O que será necessário

- Papel e caneta em todas as aulas
- Acesso a um computador
- Um **editor de texto** simples (vim, emacs, gedit, kyle, etc.)
- Um **compilador**, que transforma o código escrito em C em um programa executável (gcc)
- Capacidade para resolver problemas técnicos: como todo bom desenvolvedor, será muito importante você encontrar soluções para problemas técnicos
 - Aprendam a usar o Google

Para ir bem neste curso:

- Faça todos os laboratórios
- Faça e implemente as listas de exercícios
- E, finalmente, faça e implemente as listas de exercícios