

# Programação Competitiva IFB: Módulo 01

Prof. Daniel Saad Nogueira Nunes

13 de maio de 2022



# Sumário

- 1 Introdução
- 2 Programação Competitiva
- 3 Programação Competitiva IFB
- 4 Divulgação



# Sumário

## 1 Introdução



# Introdução

- Programação Competitiva: é um *mind sport* que visa colocar os participantes para resolver problemas através de técnicas de programação, algoritmos e estruturas de dados.
- É uma prática bem vista por diversas empresas ao redor do mundo: Google, Facebook, Amazon, . . .
- Com a prática de Programação Competitiva, o participante adquire:
  - ▶ Proficiência em paradigmas de projeto de algoritmos e estrutura de dados.
  - ▶ Capacidade de analisar a dificuldade de um problema.
  - ▶ Trabalho em equipe.
  - ▶ Conhecimento acerca de linguagens de programação.



# Introdução

- Algumas competências exercitadas:
  - ▶ Estruturas de dados.
  - ▶ Projeto por divisão e conquista.
  - ▶ Programação dinâmica.
  - ▶ Algoritmos gulosos.
  - ▶ Problemas NP-completos e NP-difíceis.
  - ▶ Busca completa.
  - ▶ Teoria dos Grafos.
  - ▶ Teoria dos Jogos.
  - ▶ Geometria Computacional.
  - ▶ Teoria dos Números.
  - ▶ Autômatos.
  - ▶ Análise de Algoritmos.
  - ▶ ...



# Programação Competitiva

- Você sabe como as big techs contratam os seus engenheiros de software?



# Maratona de Programação

## Maratona SBC

- A competição mais famosa no Brasil de programação competitiva é a **Maratona de Programação**.
- Estudantes de graduação de todo o Brasil todo participam buscando uma classificação para o ICPC (International Collegiate Programming Contest).
- Equipes de três pessoas.
- Atualmente: em torno 700 equipes participando em 50 sedes.<sup>a</sup>
- Dividida em duas fases: regional e nacional.

---

<sup>a</sup>Dados referentes à edição de 2019



# Maratona de Programação





# Olimpíada Brasileira de Informática

## OBI

- Para alunos que estejam no ensino fundamental, médio ou primeiro ano da graduação, também existe a **Olimpíada Brasileira de Informática**.
- Competição individual.
- Modalidade Programação Sênior: **Inscrições abertas até 10/06/2022.**
- Os alunos do primeiro que participam da OBI também podem participar da Maratona SBC, não são eventos excludentes.



# Olimpíada Brasileira de Informática



# Programação Competitiva IFB

- Precisamos capacitar os nossos alunos a participar da Maratona de Programação e OBI.
- Maratona de Programação todo semestre: <https://danielsaad.com/maratona/competicoes-anteriores>.
- Projeto de Ensino: Programação Competitiva IFB.



# Sumário

## 2 Programação Competitiva



# Programação Competitiva

- Para treinar para competições e aperfeiçoar o conhecimento nas diversas competências, podemos usar os **juízes online**.
- São plataformas computacionais que oferecem diversos problemas e que aceitam submissão de códigos.
- O juiz então compila o código e testa ele contra uma porção de casos de testes.
- Se seu código passa em todos os casos de teste, você ganha um **Accept**.



# Programação Competitiva

- Alguns juízes online:
  - ▶ <https://beecrowd.com.br>
  - ▶ <https://codeforces.com>
  - ▶ <https://csacademy.com>



# Programação Competitiva

- Todas essas plataformas oferecem um espaço para discussão sobre problemas.
- Desta forma, se você “empaca” em algum problema, pode pesquisar e discutir com os demais colegas.
- Ambiente totalmente favorável ao aprendizado.



# Programação Competitiva

- Vamos tomar um problema para ilustrar alguns conceitos de um juiz online.

► Beecrowd 1542





# Beecrowd 1542

## Contextualização

Você começou a competir com seu amigo para ver quem consegue ler mais livros em menos tempo. Seu amigo lia muito mais que você, até o dia que você percebeu que ele lia somente livros muito finos.

Então você resolveu contar as páginas dos livros, aumentando também a quantidade de páginas lidas por dia. Agora você lê 5 páginas por dia e termina 16 dias antes do que se estivesse lendo 3 páginas por dia. Neste cenário, quantas páginas tem o livro?



# Beecrowd 1542

## Entrada

A entrada é composta de vários casos de testes. Cada caso de teste é composto de três números  $Q$  ( $0 < Q < 20$ ),  $D$  ( $0 < D < 20$ ) e  $P$  ( $0 < P < 20$ ) separados por um espaço. Sendo que  $Q$  é a quantidade de páginas lidas por dia.  $D$  é o número de dias que você adiantaria a leitura caso estivesse lendo a quantidade de páginas informada pelo número  $P$ . Um único valor zero indica o fim da entrada.



# Beecrowd 1542

## Saída

Para cada caso de teste deverá ser impresso a quantidade de páginas do livro. (Utilize o plural corretamente e não use acentos). Este número deverá ser um inteiro, o qual representa a quantidade de páginas. Este valor deverá ser truncado caso necessário.



# Beecrowd 1542: Exemplos

## Exemplos

- Entrada:

3 16 15

0

- Saída:

120 paginas



## Beecrowd 1542: Solução

- Sabemos que

$$x = nQ$$

em que  $x$  é o número de páginas e  $n$  representa o número de dias gastos.

- Analogamente, sabemos que

$$x = \frac{(n - D)P}{P}$$



## Beecrowd 1542: Solução

- Igualando as duas equações, temos:

$$nQ = (n - D)P$$

$$nQ = nP - DP$$

$$DP = nP - nQ$$

$$n = \frac{DP}{P - Q}$$

$$x = nQ = \frac{DPQ}{P - Q}$$



## Beecrowd 1452: Solução

```
#include <stdio.h>

int main(void){
    int q,d,p;
    while(scanf("%d",&q) && q>0){
        scanf("%d %d",&d,&p);
        int n = (d*p*q/(p-q));
        if(n>1){
            printf("%d paginas\n",n);
        }
        else{
            printf("%d pagina\n",n);
        }
    }
    return 0;
}
```



## Beecrowd 1542: Solução

- Repare que, não foi impresso nada que o problema não tenha pedido.
- Evite colocar coisas do tipo: “Digite o número P”.
- A entrada é separada da leitura. Você não precisa imprimir as respostas só depois que ler toda a entrada.
- Fique atento para a condição de parada.
- Fique atento para os limites do problema!





## Juiz Online: Vereditos

Durante a sua submissão, além do **Accept**, você pode receber outros vereditos, como:



## Juiz Online: Vereditos

Durante a sua submissão, além do **Accept**, você pode receber outros vereditos, como:

- **Time Limit Exceeded (TLE)**: o seu código não conseguiu rodar dentro do tempo limite do problema. Provavelmente seu algoritmo está ineficiente.



## Juiz Online: Vereditos

Durante a sua submissão, além do **Accept**, você pode receber outros vereditos, como:

- **Time Limit Exceeded (TLE)**: o seu código não conseguiu rodar dentro do tempo limite do problema. Provavelmente seu algoritmo está ineficiente.
- **Wrong Answer (WA)**: seu problema falhou em algum caso de teste.



## Juiz Online: Vereditos

Durante a sua submissão, além do **Accept**, você pode receber outros vereditos, como:

- **Time Limit Exceeded (TLE)**: o seu código não conseguiu rodar dentro do tempo limite do problema. Provavelmente seu algoritmo está ineficiente.
- **Wrong Answer (WA)**: seu problema falhou em algum caso de teste.
- **Presentation Error (PE)**: o seu programa está correto, mas ele está imprimindo espaços ou linhas a mais (ou a menos).



## Juiz Online: Vereditos

Durante a sua submissão, além do **Accept**, você pode receber outros vereditos, como:

- **Time Limit Exceeded (TLE)**: o seu código não conseguiu rodar dentro do tempo limite do problema. Provavelmente seu algoritmo está ineficiente.
- **Wrong Answer (WA)**: seu problema falhou em algum caso de teste.
- **Presentation Error (PE)**: o seu programa está correto, mas ele está imprimindo espaços ou linhas a mais (ou a menos).
- **Compilation Error (CE)**: erro de compilação. Seu programa não conseguiu ser compilado. Verifique se escolheu o compilador correto na plataforma.



## Juiz Online: Vereditos

Durante a sua submissão, além do **Accept**, você pode receber outros vereditos, como:

- **Time Limit Exceeded (TLE)**: o seu código não conseguiu rodar dentro do tempo limite do problema. Provavelmente seu algoritmo está ineficiente.
- **Wrong Answer (WA)**: seu problema falhou em algum caso de teste.
- **Presentation Error (PE)**: o seu programa está correto, mas ele está imprimindo espaços ou linhas a mais (ou a menos).
- **Compilation Error (CE)**: erro de compilação. Seu programa não conseguiu ser compilado. Verifique se escolheu o compilador correto na plataforma.
- **Runtime Error (RE)**: erro em tempo de execução. Seu programa abortou inesperadamente durante a execução.



# Dicas para Programação Competitiva

```
#include <stdio.h>

int main(void){
    int q,d,p;
    while(scanf("%d",&q) && q>0){
        scanf("%d %d",&d,&p);
        int n = (d*p*q/(p-q));
        if(n>1){
            printf("%d paginas\n",n);
        }
        else{
            printf("%d pagina\n",n);
        }
    }
    return 0;
}
```



## Beecrowd 1542: Solução

- Repare que, não foi impresso nada que o problema não tenha pedido.
- Evite colocar coisas do tipo: “Digite o número P”.
- A entrada é separada da leitura. Você não precisa imprimir as respostas só depois que ler toda a entrada.
- Fique atento para a condição de parada.
- Fique atento para os limites do problema!





## Dicas para Programação Competitiva

- Durante a sua submissão, além do **Accept**, você pode receber outros vereditos, como os abaixo.
- **Time Limit Exceeded (TLE)**: o seu código não conseguiu rodar dentro do tempo limite do problema. Provavelmente seu algoritmo está ineficiente.
- **Wrong Answer (WA)**: seu problema falhou em algum caso de teste.
- **Presentation Error (PE)**: o seu programa está correto, mas ele está imprimindo espaços ou linhas a mais (ou a menos).
- **Compilation Error (CE)**: erro de compilação. Seu programa não conseguiu ser compilado. Verifique se escolheu o compilador correto na plataforma.
- **Runtime Error (RE)**: erro em tempo de execução. Seu programa abortou inesperadamente durante a execução.



# Sumário

## 3 Programação Competitiva IFB



# Projeto de Ensino

- Para trabalhar essas competências no IFB, elaboramos um Projeto de Ensino, dividido em módulos.
- No módulo iniciante trabalharemos com os conceitos básicos de programação competitiva e utilizaremos juízes online. para exercitar estes conceitos.
- Tudo será executado de maneira remota.



# Projeto de Ensino

## Equipe

- Coordenador: Daniel Saad.
- Tutores: Caio Aguiar, Leonam Knupp e Matheus Loiola.



# Projeto de Ensino

## Metodologia

- As aulas sobre os conceitos de programação serão **assíncronas**.
- A discussão sobre as listas de exercícios e monitoria serão **síncronas**.
- Também teremos mini-torneios de programação (Contests) como parte do projeto.
- Certificação: pode ser usada em atividades complementares.



# Projeto de Ensino

<b>Data</b>	<b>Conteúdo</b>	<b>Atividades</b>
13/05/22	Introdução a Programação Competitiva e a Linguagem C	
20/05/22	Operações Lógicas e Aritméticas	Lista 01
27/05/22	Tira-Dúvidas	
03/06/22	Estruturas de Decisão	Lista 02
10/06/22	Tira-Dúvidas	OBI
17/06/22	Estruturas de Repetição	Lista 03
24/06/22	Tira-Dúvidas	
01/07/22	Contest	
08/07/22	Vetores	Lista 04
15/07/22	Tira-Dúvidas	
22/07/22	Strings	Lista 05
29/07/22	Tira-Dúvidas	
05/08/22	Contest	
12/08/22	Finalização do Projeto de Ensino	



# Sumário

## 4 Divulgação



# Maratonas DF

- <https://facebook.com/maratonalF>.
- <https://facebook.com/maratonalDF>.
- <https://danielsaad.com/maratonal>
- <https://codeforces.com/group/btcK4I5D5f>





# Programação Competitiva IFB

- Discord do projeto “Programação Competitiva IFB”:  
<https://discord.gg/bFjmq6n3G6>



# Perguntas

Perguntas?

Contato

daniel.nunes@ifb.edu.br

