

Introdução a Maratona

Laboratório de Programação Competitiva

André Breda
Rickelme Dias

O que é a maratona

Cada time tem 1 Computador e 3 pessoas

C, C++, Java, Kotlin e Python e um sistema Linux

Resolver Problemas de Programação

Maratona Facens (11 problemas)

Maratona SBC de Programação (± 13 problemas)

- Fase Regional (geralmente setembro)
- Final Brasileira (geralmente novembro)

<https://maratona.sbc.org.br/>

<https://www3.facens.br/maratona/>



Por que participar?

A maratona irá te ajudar de diversas formas, algumas são:

- **Resolução de Problemas:** Maior capacidade na solução de problemas, envolvendo diversas estratégias e conhecimentos específicos
- **Experiência:** Aprender a lidar com equipe em ambientes competitivos
- **Profissionalmente:** Melhorando seu desempenho técnico, interação com equipes e uma maior competitividade em processos seletivos
- **Networking:** Conhecer novas pessoas e empresas
- **Prêmios:** Brindes, Viagens, Comida e outros



[Olha Que Interessante é o Processo Seletivo no Google](#)

Onde participar e praticar

Existem várias competições/contests para participar, alguns exemplos:

- Competição da Universidade
- Olimpíada Brasileira de Informática - OBI
 - Estar cursando primeira graduação e estar no primeiro ano.
- Competições de outros países
- Contests e exercícios:
 - CodeForces: <https://codeforces.com/contests>
 - Beecrowd: <https://beecrowd.com.br>
 - AtCoder: <https://atcoder.jp/contests>
 - CSES: <https://cses.fi/problemset/list>
 - VJudge: <https://vjudge.net>



Como funciona a maratona

- **Quem vence?**
 - **Mais Exercícios:** Quem resolver mais exercícios vence
 - **E/Ou, Menor Tempo:** Se empatar no exercícios, vence quem fez em menor tempo.
- **Penalidade:** Errar o envio de um exercício irá te custar tempo como penalidade, ou seja, tenha certeza do que está submetendo ao júri
- **Casos de Teste:** Praticamente todos exercícios poderão ter casos de teste no exemplo, preste atenção neles e certifique de que seu código funcione não só para os exemplos, mas também para os casos de teste “escondidos”
- **O Júri é quem testa:** O júri é quem vai rodar seu código para ver se ele atende todos os requisitos, atender todos os requisitos significa passar em todos os testes

Possíveis erros na submissão de exercícios

Erros de Submissão

1. **TLE.** Time Limit Exceeded: Acontece se sua solução passar do tempo máximo determinado pelo Júri. Pode ser que caiu em algum loop ou realmente ficou mais lento que o esperado.
2. **WA.** Wrong Answer: Acontece quando você submete uma resposta errada, isso pode acontecer se você errou na formatação do texto de saída (durante o print), ou você realmente errou a lógica do exercício (mesmo tendo acertado os testes de exemplo)
3. **PE.** Presentation Error: Certos Júris possuem esse erro quando você errou na exibição da resposta, outros Júris simplesmente irão te dar um WA (Wrong Answer).

Outros possíveis erros

4. **Compilation Error:** Seu programa nem rodou, provavelmente tem algum erro no código, verifique o código
5. **Runtime Error:** Seu programa teve algum erro enquanto rodava, verifique o código

Erros humanos e comuns, fique atento!


1. **Não prestar atenção ao problema.** Isso pode levar a mal-entendidos e suposições erradas. Certifique-se de ler o problema com atenção e entender todos os requisitos. *Tenha calma.*
2. **Tentar codificar diretamente de entradas e saídas não triviais.** Algumas entradas e saídas podem ser confusas ou não fornecer informações suficientes sobre a solução. Nesses casos, é importante fazer um passo atrás e tentar entender o problema em um nível mais abstrato. *Tente imaginar casos de teste que podem falhar e quebrar sua lógica.*

Primeiro de muitos

Primeiro exercícios de muitos:

1. Cadastre-se no <https://beecrowd.com.br/>
2. O exercício abaixo deve ser resolvido utilizando C, C++, Java ou Python
3. Você pode utilizar uma IDE de sua preferência, para começar podemos utilizar algo online como:
 - a. <https://ide.geeksforgeeks.org/online-cpp-compiler>
 - b. <https://www.onlinegdb.com/>
4. Exercício para resolver junto:
 - a. <https://www.beeccrowd.com.br/judge/pt/problems/view/1154>
5. Exercícios para ser resolvido:
 - a. <https://www.beeccrowd.com.br/judge/pt/problems/view/1000>
 - b. <https://www.beeccrowd.com.br/judge/pt/problems/view/1002>
 - c. <https://www.beeccrowd.com.br/judge/pt/problems/view/1005>
 - d. <https://www.beeccrowd.com.br/judge/pt/problems/view/1016>
 - e. <https://www.beeccrowd.com.br/judge/pt/problems/view/1035>

Primeiro de muitos


 **BEE 1154**

RESOLVIDO | SUGESTÃO | FAVORITO | DESCRIÇÃO | BLOCOS | RANKING | FÓRUM | UDEBUG

INICIANTE | NÍVEL 1 | +1.3 PONTOS | TEMPO LIMITE BASE: 1 SEGUNDOS | LIMITE DE MEMÓRIA: 200 MB

beecrowd | 1154

Idades

Adaptado por Neiloi Tonin, URI  Brasil

Timelimit: 1

Faça um algoritmo para ler um número indeterminado de dados, contendo cada um, a idade de um indivíduo. O último dado, que não entrará nos cálculos, contém o valor de idade negativa. Calcular e imprimir a idade média deste grupo de indivíduos.

Entrada

A entrada contém um número indeterminado de inteiros. A entrada será encerrada quando um valor negativo for lido.

Saída

A saída contém um valor correspondente à média de idade dos indivíduos.

A média deve ser impressa com dois dígitos após o ponto decimal.

Exemplo de Entrada	Exemplo de Saída
34 56 44 23 -2	39.25

Agradecimentos a Cassio F.

PROBLEMA

1154

LINGUAGEM

C++17

SOURCE CODE

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6
7     /**
8      * Escreva a sua solução aqui
9      * Code your solution here
10     * Escriba su solución aquí
11     */
12
13     return 0;
14 }
```

Codificar, Escolher Linguagem e Enviar

CONSTRUA A SUA SOLUÇÃO E ENVIE

ENVIAR

Exercício

Uma possível solução

Uma possível solução

```
1 #include <iostream>
2 #include <iomanip> // Biblioteca c++ para auxiliar na definição de casas decimais
3
4 using namespace std;
5
6 int main() {
7     cout << fixed << setprecision(2); // Definindo casas decimais
8
9     float numero = 0, soma = 0, quantidadeTotal = 0;
10    cin >> numero;
11
12    while (numero >= 0) {
13        soma += numero;
14
15        quantidadeTotal++;
16        cin >> numero;
17    }
18
19    cout << soma/quantidadeTotal << endl;
20
21    return 0;
22 }
23 }
```

Material para estudos e referências

Sites

- CodeMarathon: <https://www.codemarathon.com.br/>
- CP-Algorithms: <https://cp-algorithms.com/>
- Neps Academy: <https://neps.academy/problems>

Livros

- Competitive Programming: <https://cpbook.net/>

Canal do Youtube

- Carla Quem Disse: <https://www.youtube.com/@CarlaNegriLintzmayer>
- Professor Mario: <https://www.youtube.com/@ProfessorMario>
- MaratonUSP: <https://www.youtube.com/@MaratonUSP>

