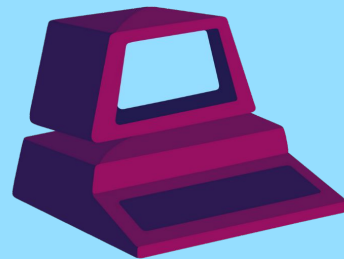


# Aula 0 - Introdução

- Maratonas, Algoritmos,  
Python vs. C++

PET



COMPUTAÇÃO

---

2023

**1.**

**Maratonas**

# OBI

---

- Prova organizada pela Unicamp;
- Do fundamental até o primeiro ano da graduação;
- Prova Individual, 2 horas
- Calouros: Modalidade Programação Nível Sênior;
- Fases: Local, Estadual, Nacional;
- Linguagens permitidas: C, C++, Python, Java, Javascript;
- Regulamento 2023: <https://olimpiada.ic.unicamp.br/info/regulamento/>
- Inscrições: algum dia

# OBI - A Prova

- Pontos de 0 a 100;
- Número de questões pode variar, com pesos diferentes;
- Existe nota parcial
- Provas dos anos anteriores: <https://olimpiada.ic.unicamp.br/passadas/>

# MFP - Maratona Feminina de Programação

- Para mulheres e pessoas não binárias;
- 4 horas e 10 problemas;
- Primeira Fase online (13/04); Fase Final na Unicamp (15/06);
- Inscrições previstas para fim de março/início de abril
- Veja a competição do ano passado:

<https://codeforces.com/group/WYIydkIPyE/contest/450037/>

# Maratona SBC de Programação

- Prova organizada pela SBC;
- Alunos de graduação e início da pós-graduação;
- Prova em Equipe de 3, com 5 horas de duração;
- Número de questões variadas, com mesmo peso;
- **Não há nota parcial**; ou acertou todos os testes ou não ganha ponto;
- Penalidade por tempo;
- Mais informações: <https://maratona.sbc.org.br/index.html>
- Ainda mais informações: <https://www.inf.ufpr.br/maratona/>

2.

Algoritmo e Programação

# O que é um algoritmo?





# O que é um algoritmo?

---

- Sequência de instruções/comandos precisos para resolver um problema;
- Geralmente, desejamos que o algoritmo seja capaz de resolver várias instâncias do mesmo problema!
- Mais formalmente, o conceito de algoritmo é ligado ao conceito de Máquina de Turing;

# O que é um algoritmo?

As linguagens de programação são equivalentes a uma Máquina de Turing: todas elas são capazes de executar qualquer algoritmo!

```
#include <stdio.h>

int main ()
{
    int x;
    scanf("%d", &x);
    for (int i = 2; i*i <= x; i++) {
        if (x % i == 0) {
            printf("Not prime\n");
            return 0;
        }
    }
    printf("Is prime\n");
    return 0;
}
```

```
x = io.read("*n")

i = 2
is_prime = true
while i*i <= x do
    if x % i == 0 then
        is_prime = false
    end
    i = i + 1
end

if is_prime then
    print("Is prime")
else
    print("Not prime")
end
```

```
x = int(input())
is_prime = True

for i in range(2, x):
    if i*i > x:
        break
    if x % i == 0:
        is_prime = False
        break

if is_prime:
    print("Is prime")
else:
    print("Not prime")
```

# O que é um algoritmo?

Então como escolhemos uma linguagem de programação?

- Velocidade: algumas linguagens tendem a ser mais rápidas que outras;
- Facilidade: algumas linguagens são mais fáceis de programar (menor tempo de desenvolvimento);
- Interação: certas linguagens podem ser mais relevantes dependendo do hardware ou do sistema em questão;

Na maratona, o que é relevante são os dois primeiros.

# C++

---

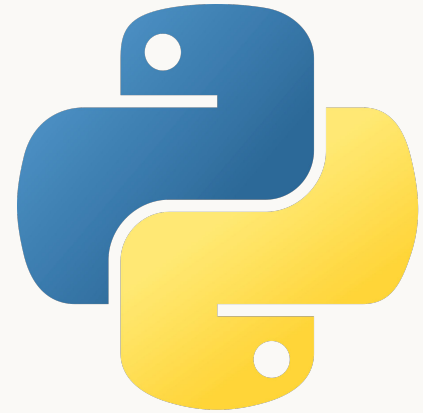
- Velocidade;
- Orientação a objetos;
- Várias estruturas úteis prontas;
- (Relativamente) fácil de programar



# Python

---

- Fácil de programar;
- Programas curtos;
- Várias estruturas úteis prontas;
- Linguagem lenta!!!



3.

Problema da OBI

[https://olimpiada.ic.unicamp.br/static/extras/obi2022/provas/ProvaOBI2022\\_f1ps.pdf](https://olimpiada.ic.unicamp.br/static/extras/obi2022/provas/ProvaOBI2022_f1ps.pdf)

## Hotel

*Nome do arquivo:* `hotel.c`, `hotel.cpp`, `hotel.pas`, `hotel.java`, `hotel.js` ou `hotel.py`

O hotel da Colônia de Férias dos Professores está com uma promoção para as férias de julho. A promoção é válida para quem chegar a partir do dia 1 de julho e sair no dia 1 de agosto.

O preço da diária do hotel é menor para quem chegar mais cedo, e vai aumentando a cada dia. Mais precisamente, a promoção funciona assim:

- A diária do hotel para cada quem chegar no dia 1 é  $D$  Reais. Assim, quem chegar no dia 1 vai pagar um total de  $31 \times D$  Reais.
- A diária do hotel aumenta  $A$  reais por dia. Ou seja, a diária é  $D + A$  Reais para quem chegar no dia 2;  $D + 2 \times A$  Reais no dia 3;  $D + 3 \times A$  Reais no dia 4 e assim por diante.
- A partir do dia 16 a diária não aumenta mais.

Note que quem chegar no dia 2 vai pagar um total de  $30 \times (D + A)$  reais; quem chegar no dia 3 vai pagar um total de  $29 \times (D + 2 \times A)$  reais, e assim por diante.

Bruno gosta muito da professora Vilma, e para agradá-la quer ajudá-la a planejar suas férias, escrevendo um programa para calcular o total (em Reais) que a professora Vilma vai gastar, dependendo do dia em que chegar no hotel.

# Pseudocódigo

Se você tivesse que fazer um passo a passo para um amigo seu resolver esse problema, como seria?

```
1. Ler D, A, N
2. dias = 32 - N
2. Se N < 16, diária = D + (N - 1) * A
3. Caso contrário, diária = D + 14 * A
4. resposta = dias * diária
5. Escreva resposta
```



4.

Como estudar para maratonas

# Cronograma dos Treinos

Nos nossos treinos, iremos ter uma parte de assunto teórico e problemas práticos, que serão resolvidos em Python

- Aula 0: Introdução
- Aula 1: Matemática Condicionais
- Aula 2: Loops e Listas
- Aula 3: Listas e Exercícios
- Aula 4: Complexidade Computacional
- Aula 5: Estruturas de Dados
- Aula 6: Técnicas Importantes
- Aula 7: Estratégias

# Outros Treinos

- Clube da UTFPR: <http://cdp.dainf.ct.utfpr.edu.br/>
- Telegram do Clube: [https://t.me/cdp\\_utfprct](https://t.me/cdp_utfprct)
- Grupo dos Iniciantes: <https://t.me/+UssY8IZYL0c0YzQx>
- Grupo Geral do Capimara



# Fora dos treinos

---

- O mais importante na maratona é fazer (muitos) exercícios!
- Compensa mais você ser bom em resolver problemas usando os assuntos básicos do que saber conteúdos avançados
- Existem vários contests de treino
- Faça provas antigas da OBI
- Recomendo fortemente o AtCoder: <https://atcoder.jp/>

# Para os mais avançados

- Se você já sabe programar, recomendo ir direto pro C++;
- Faça os exercícios das aulas em C++ ao invés de Python;
- Para estudar assuntos mais avançados, veja esse livro:  
<https://cses.fi/book/book.pdf>
- Irei deixar no final de todas as aulas exercícios mais avançados

# 5.

## Exercícios

# Exercícios

- Leia esse problema: <https://codeforces.com/gym/104555/problem/A>
- Faça um pseudocódigo dele;
- Se já souber programar (condicionais e loops), tente fazer um programa que resolva esse problema em alguma linguagem
- Leia o livreto do Capimara: <https://www.inf.ufpr.br/maratona/livreto.pdf>

# Para os mais avançados

---

- Pesquise sobre a sintaxe de C++ (o handbook pode ajudar);
- Resolva o problema do slide anterior em C++;
- Dê uma olhada nos outros problemas: <https://codeforces.com/gym/104555>



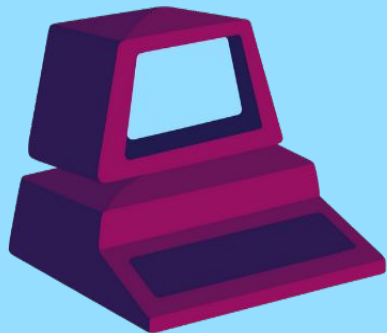
# Avalie a aula

[forms.gle/LQKytETKQ1rJA8858](https://forms.gle/LQKytETKQ1rJA8858)



Conta como presença!

# Obrigado!



**PET**  
COMPUTAÇÃO

pet.inf.ufpr.br  
pet@inf.ufpr.br  
@petcompufpr