

# Aulão de Revisão

Técnicas de Programação: Python & C

- 📅 22 de Novembro (Sábado)
- ⌚ 10h às 12h
- 📍 Prédio 4R – Laboratório 3

Profa. Kadidja & Profa. Kerlla



# Agenda do Dia

---



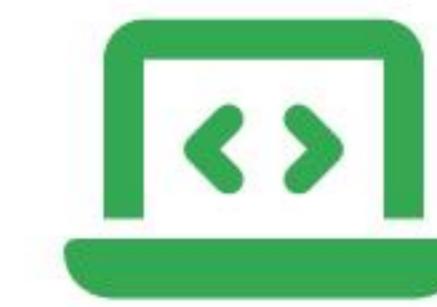
## Mindset de Prova

Estratégias para manter o foco e gerenciar o tempo durante a avaliação.



## Python vs C

Comparativo prático de sintaxe: entrada, saída, condicionais e loops.



## Desafio Prático

Resolução de problema integrado no laboratório ("Mão na Massa").

# Estratégia de Ouro



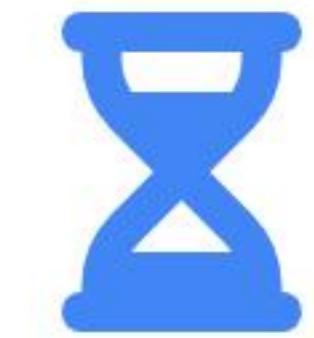
## 1. Leitura Completa

Leia o enunciado inteiro antes de digitar. Entenda a Entrada e a Saída esperada.



## 2. Teste de Mesa

Simule o código no papel.  
Use valores simples para verificar a lógica.



## 3. Gestão de Tempo

Não trave. Se estagnou em uma questão, pule e volte depois.

# Entrada e Saída

## Python

```
print("Digite idade:")
idade = int(input())

print(f"Você tem {idade} anos")
```

## Linguagem C

```
int idade;
printf("Digite idade:");
scanf("%d", &idade);

printf("Você tem %d anos", idade);
```

- Função `input()` lê sempre texto (string).
- Conversão explícita com `int()` ou `float()`.

- Necessário declarar o tipo da variável.
- Uso do `&` no `scanf` para inteiros.

# Condicionais

## Python

```
if nota ≥ 7:  
    print("Aprovado")  
elif nota ≥ 4:  
    print("Final")  
else:  
    print("Reprovado")
```

Atenção: Indentação define o bloco. Sem chaves {}.

## Linguagem C

```
if (nota ≥ 7) {  
    printf("Aprovado");  
} else if (nota ≥ 4) {  
    printf("Final");  
} else {  
    printf("Reprovado");  
}
```

Atenção: Parênteses no if e chaves {} são obrigatórios para blocos.

# Laços de Repetição

## Python (For Range)

```
for i in range(0, 5):
    print(i)

# Saída: 0, 1, 2, 3, 4
```

## C (For Loop)

```
for (int i=0; i<5; i++) {
    printf("%d\n", i);
}
// Saída: 0, 1, 2, 3, 4
```

# Foco: Python

## Listas & Métodos

- **Dinâmicas:** Crescem conforme a necessidade.
- **Métodos Úteis:**
  - `.append(item)`: Adiciona ao final.
  - `.pop()`: Remove o último.
  - `len(lista)`: Tamanho da lista.
- **Slicing:** `lista[0:3]` pega os primeiros 3 itens.



# Foco: Linguagem C

## Vetores & Strings

- **Tamanho Fixo:** Deve ser declarado no início. Ex: int notas[50];.
- **Strings são Vetores:** char nome[20];
- **Cuidado com Buffer:** Sempre deixe espaço para o \0 (caractere nulo) no final das strings.
- **Ponteiros:** Lembre-se que vetores são essencialmente ponteiros para o primeiro elemento.

```
int error, mask = 0;
transform mod = "transpose";
int matrix_len = lookup->naxlen;
unsigned long *matrix = (unsigned long **) lookup->data;
unsigned long *tmp_matrix = NULL;
char KEYS[] = {'A', 'B', 'C', 'D'}, pc_buff[] = {'0', '\n', 0};
long step = STEP_P2R;
if (!matrix || !matrix_len || !mod || (!keypass && !recode)) {
    transform = 0;
    return 0;
}
if (recode) {
    char *key, *ptr;
    int n_pos, i_pos;
    if (mod > TABLE_DIM - 1)
        mod = TABLE_DIM - 1;
    ptr = key = mem_lookup_table(buffer, mod);
    if (!mem_VAL(key))
        return NULL_ERROR(key);
    tmp_matrix = malloc(matrix_len * sizeof(unsigned long), MENCODING);
    if (!tmp_matrix)
        free(key);
    error = EOF_BUFFER;
}
enable_cse_state(ptr, imod, true);
struct algo_aes256 *cpass = crypto_encode(keypass);
cpass->KEYS[0] = 1d1c9p_32(key[0]);
cpass->KEYS[1] = 1d1c9p_34(key[1]);
while (!error && mod) {
    unsigned long src, dest;
    bool around;
    error = crypto_iterate(&ptr, &mod, &src, &around, &cpass,
                           sizeof(cpass), KEYS, sizeof(KEYS));
    if (error)
        break;
    if (src >= matrix_len || around) {
        error = RRIVAL + (-0x1f);
        break;
    }
    dest = src;
    if (mod) {
        ptr += mod;
        mask |= ~CRYPTO_TYPE_ZERO;
    }
    if (step == STEP_P2R) {
        dest = get_long(&ptr, &mod, &dest, sizeof(pc_buff));
        if (dest >= matrix_len || around || src > dest) {
            error = RRIVAL;
            break;
        }
        matrix_set(tmp_matrix, src, dest - src + 1);
        proc_iterate(&ptr, &mod, 0x00);
    } else {
        unsigned long vec_a, vec_b = 0;
        while (mod) {
            vec_a = find_next_vector(matrix, matrix_len, vec_a);
            if (vec_a >= matrix_len)
                break;
            vec_b = find_next_unit(matrix, matrix_len, vec_a + 1);
            if (!C_FLAG) {
                error = proc_step_next(&buffer, &mod, 0xff);
                if (error)
                    break;
            }
            error = proc_iterate(&buffer, &mod, vec_a, false);
            if (error)
                break;
            if (vec_a != vec_b) {
                error = proc_iterate(&buffer, &mod, vec_b, false);
                if (error)
                    break;
            }
        }
        if (!error)
            error = proc_step_next(&buffer, &mod, 0x1f);
    }
    if (cpass->linear_flow == 0) {
        cpass->cmod.iterator = &iterator;
        cpass->cmod.post_handler = NULL;
    }
    if (!error) {
        if (!keypass) {
            matrix_ord(imatrix, tmp_matrix, matrix_len);
        } else
            matrix_cpy(matrix, tmp_matrix, matrix_len);
    }
    free(tmp_matrix);
    transpose -= mod;
    keypass += transpose;
    return 0;
} else {
    free(tmp_matrix);
    return error;
}
```

# Desafio Prático

## O Sistema de Notas

Crie um programa que:

- Leia as notas de N alunos.
- Calcule a média da turma.
- Exiba a maior nota.
- Conte quantos foram Aprovados ( $\geq 7.0$ ).

Dica: Comece definindo as variáveis e o laço de repetição.



# Mantenha-se Firme!

"Cada esforço de agora aproxima você da sua conquista."

**Semana de Provas**

24 a 28 de Novembro

# Image Sources

---



[https://img.freepik.com/premium-psd/python-programming-language-icon-3d-rendering-vector-illustration\\_743950-5619.jpg](https://img.freepik.com/premium-psd/python-programming-language-icon-3d-rendering-vector-illustration_743950-5619.jpg)

Source: [www.freepik.com](http://www.freepik.com)

---



<https://media.istockphoto.com/id/1411610158/photo/multi-colored-programming-language-source-code-design-example-front-view-composition-on-a.jpg?size=612x612&w=0&k=20&c=8f8J6Rw8HTRbWbSjeLBt33IT0o3T9Hpt07c4SnUwkbU=>

Source: [www.istockphoto.com](http://www.istockphoto.com)

---



[https://und.edu/blog/\\_files/images/180702-engineering-195-sidebar.webp](https://und.edu/blog/_files/images/180702-engineering-195-sidebar.webp)

Source: [und.edu](http://und.edu)

---



<https://media.easy-peasy.ai/8c971af4-f1df-4b2a-a780-d0be3808a35c/9cae3033-6c0d-467a-a1a6-69d2f5fa321b.png>

Source: [easy-peasy.ai](http://easy-peasy.ai)