

**ANO**  
**2025**



# **UNINTER**

**CADERNO DE RESPOSTAS DA**  
**ATIVIDADE PRÁTICA DE:**

**Linguagem de Programação**

**ALUNO: TIAGO DE OLIVEIRA BRAGA**  
**GABRIEL – 5123562**

**Prof. Winston Sen Lun Fung, Me.**

## PRÁTICA 01

Desenvolver um programa em linguagem C que calcule a média ponderada de um aluno com base em suas notas e determine sua situação acadêmica.

### 1. Estrutura de Dados:

- a. Crie uma struct chamada `Notas` para armazenar as seguintes informações:
  - i. Nota da APOL1 (valor inteiro entre 0 e 100)
  - ii. Nota da APOL2 (valor inteiro entre 0 e 100)
  - iii. Nota da Prova Objetiva (valor inteiro entre 0 e 100)
  - iv. Nota da Atividade Prática (valor inteiro entre 0 e 100)

### 2. Entrada de Dados:

- a. Solicite ao usuário que digite cada uma das notas, garantindo que os valores estejam dentro do intervalo de 0 a 100.

b.

Exemplo de mensagem:

```
Digite a nota da APOL1 (0-100) :
```

### 3. Cálculo da Média:

- a. Utilize os seguintes pesos para calcular a média ponderada:
  - i. APOL1: 15%
  - ii. APOL2: 15%
  - iii. Prova Objetiva: 30%
  - iv. Atividade Prática: 40%

b. Fórmula:

$$\text{Média} = (\text{APOL1} * 0.15) + (\text{APOL2} * 0.15) + (\text{Prova Objetiva} * 0.30) + (\text{Atividade Prática} * 0.40)$$

### 4. Determinação da Situação:

- a. Classifique o aluno com base na média obtida:
  - i. **Aprovado:** Média  $\geq 70$
  - ii. **Exame:**  $30 \leq \text{Média} < 70$
  - iii. **Reprovado:** Média  $< 30$

### 5. Saída do Programa:

- a. Exiba a média calculada e a situação do aluno.

Exemplo:

```
Média: 68.5  
Situação: Exame
```

### 6. Para demonstrar o funcionamento:

- a. Use os dois primeiros dígitos do seu RU como nota da APOL1.
- b. Use os dois últimos dígitos do seu RU como nota da APOL2.
- c. Escolha valores aleatórios entre 0 e 100 para a Prova Objetiva e Atividade Prática.
- d. Faça as capturas de tela, com as entradas de dados e os resultados para o seu caderno de respostas.

## I. Apresentação do código-fonte:

```
/*
*ALUNO: Tiago de Oliviera Braga Gabriel RU:5123562
*DATA DE CRIAÇÃO:27.05.2025
*REPOSITÓRIO
GITHUB:https://github.com/tobgabriel/liguagemdeprogramacao-uninter
*/
#include <stdio.h>

//EXIGÊNCIA DE CÓDIGO 1 - Estrutura de Dados: Crie uma struct
chamada `Notas`...
typedef struct{
    //Atributos
    int notaAPOL1;
    int notaAPOL2;
    int notaProvaObjetiva;
    int notaProvaPratica;
}NOTAS;

/*EXIGÊNCIA DE CÓDIGO 2- Solicite ao usuário que digite cada uma
das notas,
garantindo que os valores estejam dentro do intervalo de 0 a
100.*/
void validaNota(int seletor, NOTAS *dado){
    int escape=1,bufferDeEntrada;
    do{
        switch(seletor){
            case 1:
                printf("\nDigite o valor da nota de APOL1(0-
100):");
                break;
            case 2:
                printf("\nDigite o valor da nota de APOL2(0-
100):");
                break;
            case 3:
                printf("\nDigite o valor da nota da prova
objetiva(0-100):");
                break;
            case 4:
                printf("\nDigite o valor da nota da prova
pratica(0-100):");
                break;
            default:
```

```
        printf("ERRO - O seletor da função validaNota está  
fora da faixa");  
        break;  
    }  
    scanf("%d",&bufferDeEntrada);  
    if(bufferDeEntrada>=0 && bufferDeEntrada<=100){  
        switch(seletor){  
            case 1:  
                dado->notaAPOL1=bufferDeEntrada;  
                break;  
            case 2:  
                dado->notaAPOL2=bufferDeEntrada;  
                break;  
            case 3:  
                dado->notaProvaObjetiva=bufferDeEntrada;  
                break;  
            case 4:  
                dado->notaProvaPratica=bufferDeEntrada;  
                break;  
            default:  
                printf("ERRO - O seletor da função validaNota  
está fora da faixa");  
                break;  
        }  
        escape=0;  
    }else{  
        printf("\nERRO - Valor de nota fora da faixa válida(0  
- 100)");  
    }  
    }while(escape);  
}  
  
//EXIGÊNCIA DE CÓDIGO 3 - Cálculo da Média: ...pesos para calcular  
a média ponderada.  
int calculaMedia(NOTAS *dado,int *pesos){  
    int index;  
    return (dado->notaAPOL1*pesos[0]+dado->notaAPOL2*pesos[1]+dado->notaProvaObjetiva*pesos[2]+dado->notaProvaPratica*pesos[3])  
        / (pesos[0]+pesos[1]+pesos[2]+pesos[3]);  
}  
  
//FUNÇÃO PRINCIPAL DO PROGRAMA  
int main(int argc, char *argv[]){
```

```
//CABEÇALHO DE IDENTIFICAÇÃO DO EXERCÍCIO
printf("\n*****
*****");
printf("\n|\tATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO\t|");
printf("\n*****
*****");
printf("\n|ALUNO:Tiago de Oliveira Braga Gabriel\t\t\t|");
printf("\n|RU:5123562\t\t\t\t\t\t|");
printf("\n*****
*****");
printf("\n|\t\t\tEXERCICIO 1\t\t\t\t|");
printf("\n*****
*****");
//VARIÁVEIS DE PROGRAMA
NOTAS aluno;
int indiceSeletor,media;
int pesoNotas[4]={15,15,30,40};
//CHAMA A FUNÇÃO PARA PREENCHER E VALIDAR TODAS AS NOTAS
for(indiceSeletor=1;indiceSeletor<=4;indiceSeletor++){
    validaNota(indiceSeletor,&aluno);
}
//CABEÇALHO SE SAÍDA DE DADOS
printf("\n*****
*****");
printf("\n|\t\t\tSAÍDA DE CÓDIGO\t\t\t\t|");
printf("\n*****
*****");
//APRESENTA AS NOTAS DIGITADAS
printf("\n\nAPOL1:%d\tAPOL2:%d\tObjetiva:%d\tPratica:%d",aluno
.notaAPOL1,aluno.notaAPOL2,aluno.notaProvaObjetiva,aluno.notaProva
Pratica);
//CALCULA A MÉDIA
media=calculaMedia(&aluno,pesoNotas);
printf("\nMédia Calculada:%d",media);
///EXIGÊNCIA DE CÓDIGO 4-Determinação da Situação :Classifique
o aluno com base na média obtida
if(media>=70){
    printf("\nSituação: Aprovado");
}else if(media<70 && media>=30){
    printf("\nSituação: Exame");
}else if(media<30){
    printf("Situação: Reprovado");
}
printf("\n\n");
```

```
return 0;  
}
```

**II. Apresentar a captura de tela evidenciando o funcionamento:**

**Para demonstrar o funcionamento:**

- a. Use os dois primeiros dígitos do seu RU como nota da APOL1.**
- b. Use os dois últimos dígitos do seu RU como nota da APOL2.**
- c. Escolha valores aleatórios entre 0 e 100 para a Prova Objetiva e Atividade Prática.**
- d. Faça as capturas de tela, com as entradas de dados e os resultados para o seu caderno de respostas.**

Figura 1 Demonstração de funcionamento atividade 1

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

agemdeprogramacao-uninter> & 'c:\Users\Usuário\.vscode\exten
DebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ksdwwhm.3v
ft-MIEngine-Error-uot3qkqh.zp4' '--pid=Microsoft-MIEngine-Pid
er=mi'

*****
|          ATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO          |
*****
|ALUNO:Tiago de Oliveira Braga Gabriel                          |
|RU:5123562                                                       |
*****
|          EXERCICIO 1                                           |
*****
Digite o valor da nota de APOL1(0-100):51

Digite o valor da nota de APOL2(0-100):62

Digite o valor da nota da prova objetiva(0-100):84

Digite o valor da nota da prova pratica(0-100):61

*****
|          SAÍDA DE CÓDIGO                                       |
*****

APOL1:51          APOL2:62          Objetiva:84          Pratica:61
Média Calculada:66
Situação: Exame

```

## PRÁTICA 02

Desenvolver um programa em linguagem C que analise um vetor de caracteres contendo o RU e o nome completo do usuário e determine a quantidade de vogais, consoantes e espaços em branco utilizando um ponteiro.

1. Estrutura de Dados:
  - a. Crie um vetor de caracteres com 120 posições para armazenar o RU e o nome completo do usuário.
2. Entrada de Dados:
  - a. Solicite ao usuário que digite seu RU e seu nome completo, garantindo que o texto digitado caiba no vetor.

Exemplo de mensagem:

```
Digite seu RU e nome completo:
```

3. Processamento:
  - a. Utilize um ponteiro para percorrer o vetor.
  - b. Verifique e conte:
    - i. Vogais: Considere as letras 'a', 'e', 'i', 'o', 'u' (maiúsculas e minúsculas).
    - ii. Consoantes: Considere todas as letras do alfabeto que não são vogais (também levando em conta letras maiúsculas e minúsculas).
    - iii. Espaços em branco: Conte os caracteres de espaço (' ').
4. Saída do Programa:
  - a. Exiba a quantidade de vogais, consoantes e espaços encontrados.

Exemplo de saída:

```
Vogais: 12
Consoantes: 18
Espaços: 3
```

5. Demonstração do Funcionamento:
  - a. Execute o programa utilizando seu RU e seu nome completo.
  - b. Faça as capturas de tela do terminal que mostrem as entradas de dados e os resultados obtidos para anexar ao seu caderno de respostas.

### III. Apresentação do código-fonte:

```
/*
*ALUNO: Tiago de Oliviera Braga Gabriel RU:5123562
*DATA DE CRIAÇÃO:27.05.2025
*REPOSITÓRIO
GITHUB:https://github.com/tobgabriel/liguagemdeprogramacao-uninter
*/
#include <stdio.h>
```



```
#include <string.h>
#include <ctype.h>
//DEFINIÇÕES DE TAMANHO
#ifndef TAMANHO_MAX
    //string deve conter 1 caractere a mais para guardar o '\0' de
    final de string
    #define TAMANHO_MAX 121
#endif

int main(int argc, char *argv[]){
    //CABEÇALHO DE IDENTIFICAÇÃO DO EXERCÍCIO
    printf("\n*****
    *****");
    printf("\n|\tATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO\t|");
    printf("\n*****
    *****");
    printf("\n|ALUNO:Tiago de Oliveira Braga Gabriel\t\t\t|");
    printf("\n|RU:5123562\t\t\t\t\t\t\t|");
    printf("\n*****
    *****");
    printf("\n|\t\t\tEXERCICIO 2\t\t\t\t|");
    printf("\n*****
    *****");
    /*EXIGÊNCIA DE CÓDIGO 1 - Estrutura de
    Dados:Crie um vetor de caracteres
    com 120 posições para armazenar o RU e o nome
    completo do usuário.*/
    char bufferDeEntrada[TAMANHO_MAX];
    int escape=1,i;
    int contadorVogais=0,contadorConsoantes=0,contadorEspacos=0;

    /*EXIGÊNCIA DE CÓDIGO 2- Entrada de Dados:Solicite ao usuário
    que digite seu RU
    e seu nome completo, garantindo que o texto digitado caiba no
    vetor.*/
    do{
        printf("\n\nDigite se RU e seu nome completo:");
        gets(bufferDeEntrada);
        //limpa o buffer do teclado
        fflush(stdin);
        //retorna valores importantes sobre o que foi inputado
        atentar par a 1 caractere a mais para armazenar '\0'
        printf("\n+-----
        ---+");
    }
```

```

printf("\n+\t\tVALORES RECEBIDOS\t\t+\n");
printf("+-----
+\n");
printf("\n->string:%s\n->tamanho:%d\n->capacidade da
string:%d\n",bufferDeEntrada,strlen(bufferDeEntrada),TAMANHO_MAX-
1);
printf("+-----
+\n");
//verifica o tamanho da string
if(strlen(bufferDeEntrada)>120){
    printf("\n\nERRO - Tamanho da string digitada acima da
capacidade.\n\n");
}else{
    //Transforma os caracteres em letras maiusculas para
facilitar o processamento
    for(i=0;i<TAMANHO_MAX;i++){
        bufferDeEntrada[i]=toupper(bufferDeEntrada[i]);
    }
    escape=0;
}
}while(escape);
/*EXIGÊNCIA DE CÓDIGO 3 - Processamento:Utilize um ponteiro
para percorrer o vetor,
verifique e conte:vogais,consoantes e espaços em branco*/
for(i=0;i<=strlen(bufferDeEntrada);i++){
    //CONTA AS VOGAIS
    if(bufferDeEntrada[i] == 'A' || bufferDeEntrada[i] == 'E'
||
        bufferDeEntrada[i] == 'I' || bufferDeEntrada[i] ==
'O' ||
        bufferDeEntrada[i] == 'U')){
        contadorVogais++;
    }
    //CONTA AS CONSOANTES
    else if((bufferDeEntrada[i]>='A' &&
bufferDeEntrada[i]<='Z') && !(bufferDeEntrada[i] == 'A' ||
bufferDeEntrada[i] == 'E' ||
        bufferDeEntrada[i] == 'I' || bufferDeEntrada[i] ==
'O' ||
        bufferDeEntrada[i] == 'U'))){
        contadorConsoantes++;
    }
    //CONTA ESPAÇOS EM BRANCO
    else if(bufferDeEntrada[i] == ' '){

```



```
        contadorEspacos++;
    }
}
//CABEÇALHO SE SAÍDA DE DADOS
printf("\n*****\n");
printf("\n|\t\t\tSAÍDA DE CÓDIGO\t\t\t|");
printf("\n*****\n");
printf("\nQuantidade de vogais:%d",contadorVogais);
printf("\nQuantidade de consoantes:%d",contadorConsoantes);
printf("\nQuantidade de espaços:%d",contadorEspacos);
return 0;
}
```

#### IV. Apresentar a captura de tela evidenciando o funcionamento:

##### *Demonstração do Funcionamento:*

- a. Execute o programa utilizando seu RU e seu nome completo.
- b. Faça as capturas de tela do terminal que mostrem as entradas de dados e os resultados obtidos para anexar ao seu caderno de respostas.

Figura 2 Demonstração de funcionamento atividade prática 2

```

*****
|          ATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO          |
|*****|
|ALUNO:Tiago de Oliveira Braga Gabriel|
|RU:5123562|
|*****|
|                      EXERCICIO 2                      |
|*****|

Digite se RU e seu nome completo:Tiago de Oliveira Braga Gabriel 5123562

+-----+
+          VALORES RECEBIDOS          +
+-----+

->string:Tiago de Oliveira Braga Gabriel 5123562
->tamanho:39
->capacidade da string:120
+-----+

*****
|                      SAÍDA DE CÓDIGO                      |
|*****|
Quantidade de vogais:14
Quantidade de consoantes:13
Quantidade de espaços:5

```

## PRÁTICA 03

Desenvolver um programa em linguagem C que utilize uma função recursiva para calcular a resistência equivalente de um conjunto de resistores conectados em paralelo. Para resistores em paralelo, a resistência equivalente é dada por:

$$R_{eq} = \frac{1}{\sum_{i=1}^n \frac{1}{R_i}}$$

### 1. Entrada de Dados:

- Crie um vetor do com o tamanho igual a quantidade de dígitos do seu RU.
- Para cada resistor, solicite o valor da resistência (em ohms).

*Exemplo de mensagens:*

```
Digite a quantidade de resistores:
Digite o valor da resistência do resistor 1 (ohms):
Digite o valor da resistência do resistor 2 (ohms):
```

### 2. Processamento:

- Armazene os valores das resistências em um vetor, o tamanho do vetor deve ser a quantidade de dígitos do seu RU.
- Implemente uma **função recursiva** que calcule a soma dos inversos das resistências:
  - **Protótipo da função:**  
`float somaInversos(float resistores[], int indice, int n);`
  - **Caso Base:** Se `indice == n`, retorne 0.
  - **Passo Recursivo:** Retorne `1/resistores[indice] + somaInversos(resistores, indice + 1, n)`.
- Após calcular a soma dos inversos, determine a resistência equivalente:

$$R_{eq} = \frac{1}{soma\ dos\ inversos}$$

### 3. Saída do Programa:

- Exiba a resistência equivalente do circuito paralelo.

*Exemplo de saída:*

```
Resistência equivalente do circuito paralelo: 5.23 ohms
```

### 4. Demonstração do Funcionamento:

- Execute o programa onde cada resistor é um dígito do seu RU, caso o dígito seja zero, utilizar o valor do dígito anterior.
- Faça capturas de tela do terminal mostrando as entradas e os resultados obtidos para anexar ao seu caderno de respostas.

### V. Apresentação do código-fonte:

```
/*
*ALUNO: Tiago de Oliviera Braga Gabriel RU:5123562
```

```

*DATA DE CRIAÇÃO:27.05.2025
*REPOSITÓRIO
GITHUB:https://github.com/tobgabriel/liguagemdeprogramacao-uninter
*/
#ifndef TAMANHO_MAX
#define TAMANHO_MAX 7
#endif
#include <stdio.h>

/*EXIGÊNCIA DE CÓDIGO 2 - PROCESSAMENTO - Implemente uma função
recursiva
que calcule a soma dos inversos das resistências: o Protótipo da
função:
float somaInversos(float resistores[], int indice, int n);
*/
float somaInversos(float resistores[],int indice,int n){
    //se chegou no ultimo resistor some 0
    if(indice == n){
        return 0;
    }else{
        return 1/resistores[indice] +
somaInversos(resistores,indice+1,n);
    }
}
int main(int argc, char *argv[]){
    //CABEÇALHO DE IDENTIFICAÇÃO DO EXERCÍCIO
    printf("\n*****");
    printf("\n|\tATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO\t|");
    printf("\n*****");
    printf("\n|ALUNO:Tiago de Oliveira Braga Gabriel\t\t\t|");
    printf("\n|RU:5123562\t\t\t\t\t\t\t|");
    printf("\n*****");
    printf("\n|\t\t\tEXERCICIO 3\t\t\t\t\t|");
    printf("\n*****");

    /*EXIGÊNCIA DE CÓDIGO 1 - Entrada de dados : Crie um vetor do
com o tamanho igual a quantidade de dígitos do seu RU.
Para cada resistor, solicite o valor da resistência (em ohms).
*/
    float resistencia[TAMANHO_MAX];

```

```

int index=0;
/*EXIGÊNCIA DE CÓDIGO 2 - Processamento: Armazene os valores
das resistências em um vetor,
o tamanho do vetor deve ser a quantidade de dígitos do seu
RU.*/
for (index=0;index<TAMANHO_MAX;index++){
    printf("\nDigite o valor da resistencia[%d]:",index+1);
    scanf("%f",&resistencia[index]);
}

//Retorna valores recebidos para computação
printf("\n+-----+");
+");
printf("\n+ \t \t VALORES RECEBIDOS \t \t + \n");
printf("+-----+");
+ \n");
for(index=0;index<TAMANHO_MAX;index++){
    printf("\n->resistencia[%d] =
%.2f",index+1,resistencia[index]);
}

//CABEÇALHO SE SAÍDA DE DADOS
printf("\n*****");
printf("\n| \t \t \t SAÍDA DE CÓDIGO \t \t \t |");
printf("\n*****");
printf("\nA soma
e: %.4f",1/somaInversos(resistencia,0,TAMANHO_MAX));
return 0;
}

```

## VI. Apresentar a captura de tela evidenciando o funcionamento:

### Demonstração do Funcionamento:

- Execute o programa onde cada resistor é um dígito do seu RU, caso o dígito seja zero, utilizar o valor do dígito anterior.
- Faça capturas de tela do terminal mostrando as entradas e os resultados obtidos para anexar ao seu caderno de respostas.

Figura 3 Demonstração de funcionamento pratica 3 parte 1

```

*****
|          ATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO          |
|*****|
|ALUNO:Tiago de Oliveira Braga Gabriel|
|RU:5123562|
|*****|
|          EXERCICIO 3          |
|*****|
Diginte o valor da resistencia[1]:5

Diginte o valor da resistencia[2]:1

Diginte o valor da resistencia[3]:2

Diginte o valor da resistencia[4]:3

Diginte o valor da resistencia[5]:5

Diginte o valor da resistencia[6]:6

Diginte o valor da resistencia[7]:2

+-----+
+          VALORES RECEBIDOS          +
+-----+

->resistencia[1] = 5.00
->resistencia[2] = 1.00
->resistencia[3] = 2.00

```



Figura 4 Demonstração de funcionamento pratica 3 parte 2

```

Diginte o valor da resistencia[1]:5
Diginte o valor da resistencia[2]:1
Diginte o valor da resistencia[3]:2
Diginte o valor da resistencia[4]:3
Diginte o valor da resistencia[5]:5
Diginte o valor da resistencia[6]:6
Diginte o valor da resistencia[7]:2

+-----+
+                VALORES RECEBIDOS                +
+-----+

->resistencia[1] = 5.00
->resistencia[2] = 1.00
->resistencia[3] = 2.00
->resistencia[4] = 3.00
->resistencia[5] = 5.00
->resistencia[6] = 6.00
->resistencia[7] = 2.00
*****
|                SAÍDA DE CÓDIGO                |
*****
A soma e:0.3448

```

## PRÁTICA 04

Desenvolver um programa em linguagem C que calcule a quantidade mínima de lâmpadas necessárias para iluminar um cômodo, utilizando a norma NBR 5413. O programa deverá solicitar apenas a largura, o comprimento e o tipo do cômodo.

### 1. Entrada de Dados:

- ✓ Solicite ao usuário que informe:
  - Tipo do cômodo:  
Selecione uma das opções abaixo:  
1 - Quarto  
2 - Escritório  
3 - Cozinha
- ✓ Comprimento do cômodo (em metros)
- ✓ Largura do cômodo (em metros)
- ✓ Comprimento do cômodo (em metros)

Exemplo de mensagens:

```
Selecione o tipo de cômodo:
1 - Quarto
2 - Escritório
3 - Cozinha
Digite a opção desejada:

Digite o comprimento do cômodo (m):
Digite a largura do cômodo (m):
```

### 2. Processamento:

- ✓ Determinar a Iluminância Recomendada:

Utilize a seguinte tabela, baseada na NBR 5413:

Tipo de Cômodo	Iluminância Recomendada (lux)
1 - Quarto	150
2 - Escritório	300
3 - Cozinha	300

- ✓ Cálculo da Área do Cômodo:  
$$\text{Área} = \text{largura} \times \text{comprimento}$$
- ✓ Parâmetros Fixos do Sistema:  
Para simplificar, considere os seguintes valores:
  - Fluxo luminoso de cada lâmpada: 800 lúmens
  - Fator de manutenção: 0.8
- ✓ Cálculo do Fluxo Luminoso Total Necessário:  
Para atingir a iluminância recomendada, o fluxo total necessário (em lúmens) é dado por:  
$$\text{Fluxo Total} = \frac{\text{Área} \times \text{Iluminância Recomendada}}{\text{Fator de Manutenção}}$$
- ✓ Determinar a Quantidade Mínima de Lâmpadas:  
Divida o fluxo total necessário pelo fluxo luminoso de cada lâmpada:

$$\text{Número de Lâmpadas} = \frac{\text{Fluxo Total}}{800}$$

Caso o resultado não seja um número inteiro, arredonde-o para cima (utilize a função `ceil()` da biblioteca `<math.h>`).

- ✓ Saída do Programa:
  - Exiba ao usuário:
    - O tipo de cômodo selecionado e a iluminância recomendada correspondente.
    - A área calculada do cômodo.
    - O fluxo luminoso total necessário.
    - A quantidade mínima de lâmpadas recomendadas.

Exemplo de saída:

```
Tipo de cômodo: Escritório (Iluminância recomendada: 300 lux)
Área do cômodo: 20.00 m²
Fluxo luminoso total necessário: 7500.00 lúmens
Quantidade mínima de lâmpadas recomendadas: 10
```

#### 4. Demonstração do Funcionamento:

- ✓ Execute o programa informando como largura a soma dos dois primeiros dígitos do seu RU e para o comprimento os dois últimos dígitos do seu RU. **Atenção: caso a soma resulte em ZERO utilize o valor 12.3 .**
- ✓ Faça capturas de tela do terminal exibindo as entradas e os resultados obtidos, e anexe ao seu caderno de respostas.

#### Dicas para Implementação:

- ✓ Utilize as bibliotecas `<stdio.h>` para entrada e saída de dados e `<math.h>` para a função `ceil()`.
- ✓ Valide a opção selecionada para o tipo de cômodo, garantindo que seja 1, 2 ou 3.
  - Certifique-se de realizar os cálculos com precisão (use o tipo `float` ou `double` conforme necessário).

#### VII. Apresentação do código-fonte:

```
/*
*ALUNO: Tiago de Oliviera Braga Gabriel RU:5123562
*DATA DE CRIAÇÃO:03.06.2025
*REPOSITÓRIO
GITHUB:https://github.com/tobgabriel/liguagemdeprogramacao-uninter
*/
#include <stdio.h>
#include <math.h>

int main(int argc, char *argv[]){
    //VARIÁVEIS DE SISTEMA
    int escape=1;
    int tipoComodo=0,numeroLampadas;
    float comprimento,largura,area,fluxoTotal;
```

```
const float
tabelaIluminancia[3]={150.0,300.0,300.0},fatorDeManutencao=0.8;
//CABEÇALHO DE IDENTIFICAÇÃO DO EXERCÍCIO
printf("\n*****
*****");
printf("\n|\tATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO\t|");
printf("\n*****
*****");
printf("\n|ALUNO:Tiago de Oliveira Braga Gabriel\t\t\t|");
printf("\n|RU:5123562\t\t\t\t\t\t\t|");
printf("\n*****
*****");
printf("\n|\t\t\tEXERCICIO 3\t\t\t\t|");
printf("\n*****
*****");

/*EXIGÊNCIA DE CÓDIGO 1 - Entrada de dados: Solicita ao
usuário que informe:
tipo de comodo, comprimento e largura*/
while(escape){
    printf("\nSelecione o tipo de comodo:\n1-Quarto\n2-
Escritório\n3-Cozinha\n>_");
    scanf("%d",&tipoComodo);
    //VALIDA VALOR DE LEITURA
    fflush(stdin);//limpa bufer de teclado
    if(tipoComodo>=1 && tipoComodo<=3){
        escape=0;
    }else{
        printf("\nERRO - Valor digitado fora da faixa\n");
    }
}
escape=1;
while(escape){
    printf("\nValor do comprimento do comodo(metros):");
    scanf("%f",&comprimento);
    fflush(stdin);
    if(comprimento>0){
        escape=0;
    }else{
        printf("\nERRO - Valor digitado fora da faixa\n");
    }
}
escape=1;
while(escape){
```

```
printf("\nValor da largura do comodo(metros):");
scanf("%f",&largura);
fflush(stdin);
if(largura>0){
    escape=0;
}else{
    printf("\nERRO - Valor digitado fora da faixa\n");
}
}

area=comprimento*largura;
fluxoTotal=(area*tabelaIluminancia[tipoComodo])/fatorDeManutencao;
numeroLampadas=ceil(fluxoTotal/800);

//Retorna valores recebidos para computação
printf("\n+-----+");
+");
printf("\n+\t\tVALORES RECEBIDOS\t\t +\n");
printf("+-----+");
+");
printf("tipo
comodo:%d\tcomprimento:%.2f\tlargura:%.2f",tipoComodo,comprimento,
largura);

//CABEÇALHO SE SAÍDA DE DADOS
printf("\n*****
*****");
printf("\n|\t\t\tSAÍDA DE CÓDIGO\t\t\t|");
printf("\n*****
*****");
switch(tipoComodo){
    case 1:
        printf("\nComodo:Quarto\t");
        break;
    case 2:
        printf("\nComodo:Escritório\t");
        break;
    case 3:
        printf("\nComodo:Cozinha\t");
        break;
    default:
        printf("\nComodo selecionado não valido\t");
        break;
```



```
}  
printf("\nArea calculada:%.2f m²",area);  
printf("\nFluxo total necessario:%.2f lm",fluxoTotal);  
printf("\nQuantidade minima de lampadas(800 lm):%d  
und",numeroLampadas);  
  
return 0;  
}
```

#### VIII. Apresentar a captura de tela evidenciando o funcionamento:

##### Demonstração do Funcionamento:

- ✓ Execute o programa informando como largura a soma dos dois primeiros dígitos do seu RU e para o comprimento os dois últimos dígitos do seu RU. Atenção: caso a soma resulte em ZERO utilize o valor 12.3 .
- ✓ Faça capturas de tela do terminal exibindo as entradas e os resultados obtidos, e anexe ao seu caderno de respostas.

Figura 5 Demonstracao pratica 4

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

|          ATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO          |
|*****|
|ALUNO:Tiago de Oliveira Braga Gabriel|
|RU:5123562|
|*****|
|          EXERCICIO 3          |
|*****|
Selecione o tipo de comodo:
1-Quarto
2-Escritório
3-Cozinha
>_1

Valor do comprimento do comodo(metros):6

Valor da largura do comodo(metros):62

+-----+
+          VALORES RECEBIDOS          +
+-----+
tipo comodo:1   comprimento:6.00      largura:62.00
*****
|          SAÍDA DE CÓDIGO          |
|*****|
Comodo:Quarto
Area calculada:372.00 m²
Fluxo total necessario:139500.00 lm
Quantidade minima de lampadas(800 lm):175 und

```