

**ANO**  
**2023**



# **UNINTER**

## **ATIVIDADE PRÁTICA**

### **LINGUAGEM DE PROGRAMAÇÃO**

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



---

## INTRODUÇÃO

Olá a todos.

Sejam todos muito bem-vindos!

Esta avaliação foi planejada e preparada para as disciplinas de Linguagem de Programação Centro Universitário Internacional Uninter.

O objetivo desta atividade é fazer com que você, aluno, desenvolva os conhecimentos teóricos aprendidos na rota de maneira.

Ao longo desse roteiro serão passadas as orientações gerais para realização da avaliação bem como os seus critérios de correção.

*No mais, desejo-lhe boa atividade prática em nome dos professores  
da disciplina de Linguagem de Programação.*



## SUMÁRIO

<b>INTRODUÇÃO</b>	<b>1</b>
<b>ORIENTAÇÕES GERAIS</b>	<b>3</b>
<b>Atividade PRÁTICA</b>	<b>5</b>
PRÁTICA 01	5
PRÁTICA 02	6
PRÁTICA 03	7
PRÁTICA 04	8
PRÁTICA 05	9

## ORIENTAÇÕES GERAIS

Esta atividade é constituída de cinco questões, sendo que cada uma delas contribui com 20% para a nota total. Para obter a pontuação completa em cada exercício, você precisará satisfazer os três requisitos essenciais descritos nas ORIENTAÇÕES GERAIS:

1. Submissão de um código fonte completo, que deve estar devidamente indentado e organizado;
2. Inclusão de comentários explicativos em seu próprio código fonte;
3. O código fonte deve estar completamente funcional e corresponder às exigências do enunciado do exercício;
4. Incorporação de IMAGENS do terminal em ação, demonstrando as solicitações de cada exercício.

Estas práticas devem ser desenvolvidas utilizando o **VISUAL STUDIO COMMUNITY 2017, 2019 ou 2022** em Linguagem C estruturada. Códigos criados em outros compiladores não serão aceitos.

É necessário entregar um **ÚNICO ARQUIVO EM FORMATO PDF** no AVA, através do ícone Trabalhos. Este arquivo deve seguir o formato descrito no documento Caderno de Respostas, disponível na Rota de Estudos.

Você é encorajado a realizar consultas e pesquisas para a conclusão desta atividade. Recomenda-se a leitura de livros base, materiais complementares e a interação com o tutor para esclarecimento de dúvidas, através do Canal de Tutoria.

Para cada exercício, inclua no seu relatório o enunciado do exercício, além de uma resposta contendo seu código COMPLETO - desde a primeira até a última linha digitada por você. Insira também comentários em seu próprio código, explicando seu funcionamento.

Para inserir o código nas respostas, basta copiar (CTRL+C) o código que você criou no VISUAL STUDIO e colar (CTRL+V) no documento Word. O código já aparecerá organizado e indentado.

Existe um modelo em WORD o **Caderno de Respostas** no AVA para o seu uso **obrigatório**. Ao submeter seu arquivo, entretanto, converta-o para o formato PDF (opção "Exportar para PDF" no Word).

Adicionalmente, é preciso incluir uma captura de tela do seu código em funcionamento. A imagem deve mostrar o terminal executando o seu código e exibindo os dados solicitados na tela, conforme o exemplo fornecido no modelo.



Caso seu código esteja corretamente desenvolvido e seja funcional, mas esteja sem os comentários explicativos ou as imagens demonstrativas do código em funcionamento no terminal, sua nota será significativamente reduzida.

## CUIDADO!

Na área de programação, é quase impossível encontrar dois códigos que sejam exatamente idênticos. Cada programador tem seu próprio estilo de organizar o código, nomear variáveis, fazer comentários, gerar mensagens para os usuários, entre outras coisas. Portanto, não será permitida a entrega de algoritmos idênticos entre os alunos ou que sejam cópias da Internet. Se o avaliador detectar respostas idênticas, estas serão classificadas como PLÁGIO e a penalidade será a atribuição de NOTA ZERO para a questão.

Trabalhos entregues em formatos diferentes do PDF não serão corrigidos e será atribuído a nota ZERO para ele.

Trabalhos entregues sem a utilização do CADERNO DE RESPOSTAS não serão corrigidos e será atribuído a nota ZERO para ele.

## Atenção:

Imagine o RU 1 2 3 4 5 6 7

1	2	3	4	5	6	7
Primeiro dígito						Último dígito



## ATIVIDADE PRÁTICA

### PRÁTICA 01

Implemente um programa em C que faça o seguinte:

- Crie uma estrutura chamada "Estudante" que armazene um nome completo (até 100 caracteres) e o RU (7 dígitos<sup>1</sup>).
- Peça ao usuário para inserir o seu nome completo e RU.
- Crie um ponteiro que aponte para essa estrutura.
- Usando o ponteiro, verifique se o RU é par ou ímpar, imprimindo o resultado.
- Use o ponteiro para imprimir o nome do estudante ao contrário. Por exemplo, se o nome for "João da Silva", o programa deve imprimir "avliS ad oãoJ".

Para demonstrar o funcionamento informe o seu nome e o seu RU, faça as capturas de tela do terminal de execução.

Exemplo de funcionamento:

**Entrada:**

```
Nome: João da Silva  
RU: 1234567
```

**Saída:**

```
O RU é ímpar  
Nome invertido: avliS ad oãoJ
```

---

<sup>1</sup> Caso o seu RU tenha menos de 7 dígitos altere o tamanho do vetor para a quantidade de dígitos do seu RU.



## PRÁTICA 02

Escreva um programa em C que siga os seguintes passos:

- Solicite ao usuário que insira o seu RU<sup>1</sup>.
- Crie um vetor cujo tamanho seja igual ao número de dígitos no RU. Por exemplo, se o RU for "1234567", então o vetor terá 7 posições.
- Peça ao usuário para inserir cada dígito do seu RU, armazenando cada dígito em uma posição do vetor. Por exemplo, se o RU for "1234567", o vetor deverá ter os elementos [1, 2, 3, 4, 5, 6, 7].
- Usando ponteiros, calcule o produto de todos os números no vetor.

Para demonstrar o funcionamento informe o seu RU e faça as capturas de tela do terminal de execução.

Exemplo de funcionamento:

**Entrada:**

RU: 1234567

**Saída:**

Produto dos dígitos do RU: 5040

## PRÁTICA 03

Escreva um programa em C que siga os seguintes passos:

- Solicite ao usuário que insira o comprimento do lado (L) de um pentágono regular.
- Implemente uma função chamada `calc_penta` que recebe o comprimento do lado do pentágono e dois ponteiros para float. Esta função deve calcular a área e o perímetro do pentágono e armazenar os resultados nos endereços apontados pelos ponteiros. A função deve seguir o protótipo abaixo:

**`void calc_penta(float l, float *area, float *perimetro);`**

- No programa principal, depois de receber o valor do lado do usuário, chame a função `calc_penta`, passando o valor do lado e os endereços de duas variáveis de float (para armazenar a área e o perímetro calculados).
- Imprima a área e o perímetro calculados na tela.
- O programa deve continuar pedindo ao usuário um novo valor de lado e calculando a área e o perímetro até que o usuário insira um valor negativo para o lado.

Para demonstrar o funcionamento utilize informe para o valor do lado o primeiro dígito do seu RU.

Nota: A área (A) e o perímetro (P) de um pentágono regular podem ser calculados com as seguintes fórmulas:

$$A = \frac{5 \cdot L^2 \cdot \tan\left(\frac{\pi}{5}\right)}{4}$$

$$P = 5 \cdot L$$

Exemplo de funcionamento:

**Entrada:**

Lado do pentagono: 5

**Saída:**

Area do pentagono: 43.0119

Perímetro do pentagono: 25





## PRÁTICA 04

Desenvolva, em linguagem C, um programa que calcule o valor RMS (Root Mean Square) de um sinal elétrico. O valor RMS é uma medida estatística da magnitude de um sinal variável. Ele pode ser especialmente útil ao lidar com quantidades variáveis no tempo, como a corrente ou a tensão em um circuito elétrico.

Seu programa deverá:

Solicitar ao usuário para inserir o número de leituras do sinal que deseja analisar.

Receber do usuário essa sequência de leituras, uma por uma.

Utilizar uma função recursiva para calcular a **soma dos quadrados** dessas leituras.

Finalmente, calcular e imprimir o valor RMS do sinal. Lembre-se, o valor RMS é a raiz quadrada da média dos quadrados das leituras.

Para a função recursiva, você pode usar o protótipo:

**`double somaQuadrados(double leituras[], int tamanho, int indice)`**

Para demonstrar o funcionamento informe a quantidade de medidas igual a quantidade de dígitos do seu RU e para cada uma das medidas informe cada dígito do seu RU.

Exemplo de funcionamento:

**Entrada (exemplo de RU de 4 dígitos):**

```
Digite o numero de leituras: 4
Digite a leitura 1: 2
Digite a leitura 2: 3
Digite a leitura 3: 4
Digite a leitura 4: 5
```

**Saída:**

```
O valor RMS do sinal é: 3.87
```



## PRÁTICA 05

Seu objetivo é criar um programa em linguagem C para manipular e armazenar informações pessoais.

Você deve começar solicitando ao usuário que insira dados para seis diferentes registros. Cada registro deve incluir as seguintes informações: Nome, CPF, Telefone(DDD+nº do telefone) e E-mail.

Uma vez coletados, você deve armazenar esses dados em um arquivo no formato CSV. Para manter os dados organizados, use o caractere ponto e vírgula (;) como delimitador de campo.

O nome do arquivo a ser gerado deve ser o seu número de Registro Único (RU). Assim, se o seu RU for 123456, o nome do arquivo deve ser "123456.csv".

Exemplo de funcionamento:

### Entrada:

```
Digite os dados para o registro 1:  
Nome: Joao Silva  
CPF: 11111111111  
Telefone: (11) 11111111  
E-mail: joao.silva@example.com
```

```
Digite os dados para o registro 2:  
Nome: Maria Santos  
CPF: 22222222222  
Telefone: (22) 22222222  
E-mail: maria.santos@example.com
```

...

```
Digite os dados para o registro 6:  
Nome: Ana Costa  
CPF: 66666666666  
Telefone: (66) 66666666  
E-mail: ana.costa@example.com
```

### Saída (no arquivo "123456.csv"):

```
Nome;cpf;telefone;email  
Joao Silva;11111111111;11 11111111;joao.silva@example.com  
Maria Santos;22222222222;22 22222222;maria.santos@example.com  
...  
Ana Costa;66666666666;66 66666666;ana.costa@example.com
```