

ATIVIDADE PRÁTICA

1. OBJETIVO

Desenvolver os algoritmos do item 5 em **linguagem de programação C**.

2. MATERIAL UTILIZADO

A Atividade Prática de Lógica de Programação e Algoritmos será realizada com a utilização do software **VISUAL STUDIO COMMUNITY 2019**. Orientações para instalação e uso do software estão disponíveis em vídeo na ROTA DE APRENDIZAGEM (AULA 14 e MATERIAIS COMPLEMENTARES).

Caso você trabalhe com outro sistema operacional. Ou sua máquina não suporte os requisitos de configuração do Visual Studio. Recomendo que utilize uma destas duas ferramentas:

- **Visual Code**
 - Disponível também no site da Microsoft;
 - <https://code.visualstudio.com/>
- **Ferramenta Online Repl**
 - Você pode criar uma conta neste site e programar em C sem precisar instalar;
 - <https://repl.it/>

IMPORTANTE: as ferramentas extras que coloquei são sugestões. Isso significa que a única ferramenta que teremos tutorial de instalação é a oficial (Visual Studio). Além disso, observe que para realizar a prova discursiva você terá acesso somente ao Visual Studio, então é importante que você se familiarize com ele. Caso não tenha ele em casa, tente ir ao seu polo em algum momento para testá-lo e praticar.

3. ORIENTAÇÕES GERAIS

- O aluno deverá entregar um **ARQUIVO ÚNICO NO FORMATO PDF OU DOCX** no AVA no ícone **Trabalhos**;
 - Submeter o trabalho em múltiplos arquivos separados, ou em formatos diferentes dos impostos acima, acarretará em nota zero;
 - Esta atividade é para ser realizada com consulta e pesquisa. Portanto, leia também livros bases, materiais complementares ou procure seu tutor para tirar dúvidas diretamente via Canal de Tutoria;
 - Para cada exercício, coloque no seu relatório o ENUNCIADO do mesmo e coloque como resposta o seu código COMPLETO. Deste a primeira até a última linha que você digitou;
 - Coloque no seu código COMENTÁRIOS explicando COM SUAS PALAVRAS o que ele faz (veja o exemplo do documento modelo);
 - Para inserir os códigos nas respostas, faça um **CTRL+C/CTRL+V** do código criado por você no VISUAL STUDIO e cole-o no documento do Word. Assim, o código já virá **colorido**, organizado e indentado;
 - No AVA existe um modelo em WORD para você utilizar. Se você optar por submeter o seu arquivo em PDF, basta apertar em salvar como PDF no Word;
 - Além do seu algoritmo, você deverá colocar uma captura de tela do seu código funcionando. Capture o terminal mostrando o seu código funcionando e imprimindo os dados solicitados na tela (veja o exemplo do modelo).
-
- **CUIDADO!**
 - ✓ Em programação, não existem dois códigos exatamente iguais. Cada programador organiza seu código de uma forma diferente, declara variáveis com nomes diferentes, faz comentários diferentes, gera mensagens aos usuários distintas, etc. Por este motivo, não serão aceitos dois algoritmos idênticos entre alunos (ou iguais à Internet). Caso o corretor observe respostas iguais, elas serão consideradas como **PLÁGIO** e será atribuída a **NOTA ZERO** na questão;

4. COMO SE DARÁ A CORREÇÃO DA ATIVIDADE?

Como temos 3 questões. Seus pesos são:

- Exercício 1 – 25%;
- Exercício 2 – 25%;
- Exercício 3 – 50% (25% cada item);

Para que você ganhe nota máxima em cada exercício, você precisará cumprir os três requisitos básicos explicados nas ORIENTAÇÕES GERAIS:

- Apresentar seu algoritmo completo, indentado e organizado;
- Explicar seu código através de comentários;
- Colocar uma IMAGEM com o terminal rodando e mostrando o que cada exercício pede.

No modelo de relatório da disciplina você encontrará um exemplo de exercício para um melhor entendimento. Caso você desenvolva seu código corretamente e funcional, porém não faça os comentários nem coloque uma imagem dele funcionando no terminal, terá sua nota severamente prejudicada.

5. EXERCÍCIOS

Resolva os algoritmos abaixo seguindo todas as instruções listadas neste documento.

Exercício 1:

Faça um algoritmo em linguagem C para calcular quanto cada cliente gastou em pedidos realizados em uma lancheria. A lancheria vende os seguintes produtos:

Item	Produto	Código	Preço Unitário
1	Cachorro-quente	100	5,00
2	X-salada	101	8,79
3	X-bacon	102	9,99
4	Misto	103	6,89
5	Salada	104	4,80
6	Água	105	3,49
7	Refrigerante	106	4,99

Ao iniciar o programa, deve-se aparecer na tela um menu com todos os itens vendidos e seus respectivos preços. O usuário deve escolher digitando de 1-7 qual item quer comprar.

Observe que o usuário pode comprar o mesmo item diversas vezes e também pode comprar vários itens diferentes (exemplo: 2 águas e um cachorro-quente, no mesmo pedido).

Para encerrar o processo de compra, o usuário deve digitar um valor diferente de 1-7. Ao encerrar a compra, deve-se listar todos os itens comprados, seus preços individuais e o total a se pagar.

Imprima na tela um teste do seu programa utilizando os três primeiros e os últimos dígitos do seu RU da UNINTER como os valores de entrada para os pedidos. Os três

primeiros dígitos serão os itens comprados, e os três últimos a quantidade de cada item comprado. Por exemplo, se seu RU for 1234567, a compra resultante será:

Item	Produto	Quantidade
1 (1º dígito)	Cachorro-quente	5 (antepenúltimo dígito)
2 (2º dígito)	X-salada	6 (penúltimo dígito)
3 (3º dígito)	X-bacon	7 (último dígito)

Exercício 2:

Faça um algoritmo em linguagem C que receba como dado de entrada a quantidade máxima em milímetros que um reservatório de água suporta. E também como dado de entrada informe quantos milímetros de chuva tivemos em cada mês do ano, e quantos milímetros deste reservatório por mês foi gasto. Armazene as informações de cada mês em vetores de dados.

Sabendo a capacidade do reservatório, quanto choveu em cada mês e quanto foi gasto. Informe o valor disponível de água, em milímetros, para cada mês do ano. Caso o reservatório fique vazio em algum momento, informe isso na tela. Caso o reservatório fique cheio em algum momento, também informe isso na tela.

Imprima na tela um teste do seu programa utilizando o seu RU da UNINTER como os valores de entrada, cada dígito do seu RU será um dos valores gastos, em milímetros, por mês. Por exemplo, se seu RU for 1234567, faça:

Jan	1 mm (1º dígito)
Fev	2 mm (2º dígito)
Mar	3 mm (3º dígito)
Abr	4 mm (4º dígito)
Mai	5 mm (5º dígito)
Jun	6 mm (6º dígito)

Jul	7 mm (7º dígito)
Ago	1 mm (1º dígito)
Set	2 mm (2º dígito)
Out	3 mm (3º dígito)
Nov	4 mm (4º dígito)
Dez	5 mm (5º dígito)

Exercício 3:

Faça um algoritmo em linguagem C que receba como dado de entrada uma matriz de caracteres. Cada posição da matriz deve armazenar um caractere.

A dimensão da matriz deverá ser $M \times N$, onde M e N são dados de entrada escolhidos pelo usuário. Declare a matriz como sendo de dimensão máxima 10×10 , e valide os valores de M e N , que devem estar entre 1 e 10, incluindo 1 e 10. Para a validação, o usuário deve ficar preso no programa enquanto não digitar um número válido para M e para N .

Após escolhida a dimensão da matriz e povoado ela com caracteres. Faça:

- Conte o número de ocorrências de cada caractere que aparece na matriz. Após a contagem, liste cada caractere digitado e o número de ocorrências;
- Crie uma função que receba como parâmetro o primeiro caractere cadastrado na matriz. A função deve retornar um dado numérico. Se o caractere for maiúsculo, retorne na função o resultado deste caractere dividido por 10. Caso contrário, retorne o resultado do caractere multiplicado por 2;

Imprima na tela um teste do seu programa utilizando como primeiro caractere a primeira letra do seu nome.