

Curso <i>Bacharelado em Ciência da Computação</i>			Instituto <i>Instituto de Ciências Exatas e Tecnológicas</i>
Disciplina <i>AP2 – Algoritmos e Programação 2</i>			
Nome do(a) acadêmico(a)			Assinatura
Nº de matrícula	Turma <i>2º Período</i>	Data <i>09/09/2024</i>	Professor(a) <i>Ana Paula Freitas Vilela Boaventura</i>

ORIENTAÇÕES PARA A RESOLUÇÃO - O conteúdo exigido para resolução desta lista de exercícios compreende os seguintes capítulos no *Plano de Ensino* da disciplina: Passagem de parâmetros por valor.

1 – Desenvolver um programa em C que apresente um menu iterativo para o usuário, permitindo a escolha dos exercícios desta lista. Cada exercício deverá ser implementado em sub-programas, organizados em arquivos de cabeçalho (.h), e o menu será gerenciado utilizando a estrutura de controle switch case, dentro de um laço de repetição.

2 – No arquivo DOIS.h, crie uma função int POSITIVO_NEGATIVO (int n), que receba um valor e retorne 1 se o número digitado for positivo ou 0 se o número for negativo.

3 – No arquivo TRES.h, faça uma sub-rotina que receba um único valor representando segundos. Essa sub-rotina deverá convertê-lo para horas, minutos e segundos. Todas as variáveis devem se passadas como parâmetros, não havendo variáveis globais.

4 – No arquivo QUATRO.h, crie e uma sub-rotina que receba 2 números positivos por parâmetro e retorne a soma dos números existentes entre eles.

5 - No arquivo CINCO.h, crie e uma sub-rotina int TRIANGULO (int a, int b, int c), que verifique se os valores passados formam um triângulo e se formarem o triângulo, imprima na tela o tipo de triângulo.

6 – No arquivo SEIS.h, lembramos que, matematicamente, calcular as combinações possíveis é contar quantos subconjuntos podemos formar com parte dos elementos do conjunto, sem nos importarmos com a ordem. Para calcular a combinação, utilizamos a Equação 1:

$$C_{(n,p)} = \frac{n!}{p! (n - p)!} \quad (1)$$

Sendo:

n = quantidade total de elementos

p = partes dos elementos do conjunto

Na Escola de Games/BCC, a equipe é numerosa e criamos Grupos de Trabalho (GT) para desenvolvermos as ações. Sabendo que o time possui 10 integrantes e o GT é composto por 4 membros, ao aplicar a Equação 1, chegamos a impressionante marca de 210 possíveis combinações de equipe.

A Proece/UFJ reconhecendo a nossa “tecnologia”, solicitou a turma de AP2/2024 para que auxiliasse os membros de outros projetos. Portanto, a sua missão é criar uma sub-rotina que calcule quantos grupos de trabalhos distintos cada projeto será capaz de criar. Primeiramente, crie a sub-rotina FATORIAL (int valor), que será usada para calcular o numerador e denominador da Eq. 1. Depois, crie a int COMBINA (int n, int p) para apresentar o resultado final.

7 – No arquivo SETE.h, você foi contratado para automatizar uma garagem de estacionamento. A empresa cobra R\$20,00 de taxa mínima para estacionar até três horas. A garagem cobra um adicional de R\$5,00 por hora ou fração caso sejam excedidas as três horas. A taxa máxima para qualquer período determinado de 24 horas é R\$50,00. Admita que nenhum carro fique estacionado por mais de 24 horas. Escreva um programa que calcule e imprima as taxas de estacionamento para três clientes que estacionaram ontem seus carros nessa garagem. Você deve fornecer as horas que cada cliente ficou estacionado. Seu programa deve imprimir os resultados organizados em um formato de tabela e deve calcular e imprimir o total recebido no dia de ontem. O programa deve usar a função float CALCULA_TAXAS (int horas) para determinar o valor a ser cobrado de cada cliente. A saída de seu programa deve ter o seguinte formato:

Carro	Horas	Taxas
1	1	20,00
2	4	30,00
3	24	50,00
Total	25	85,00

Note que ao final, será necessário somar o total de horas e das taxas de todos os elementos do vetor.

Desafio

Créditos: Renato Franco

Escreva um programa em C que simule um sistema bancário simplificado. O programa deverá conter duas contas bancárias e permitir ao usuário realizar as seguintes operações:

Depositar uma quantia em uma das contas.

Sacar uma quantia de uma das contas.

Transferir uma quantia de uma conta para outra.

Exibir os saldos das duas contas.

As operações de depósito e saque devem ser feitas através de funções que recebem o saldo da conta como parâmetro por valor. O saldo **NÃO DEVE SER ALTERADO** dentro da função, mas a função deve **RETORNAR** o novo saldo para ser atualizado no programa principal.

Criação de bibliotecas é permitido, porém não é uma situação obrigatória. Será avaliado o seu conhecimento de passagem de parâmetros por valor, então explore a sua melhor forma de criar o programa.

O programa deverá ter tratamentos de erro comuns em contas bancárias, o que se espera de tratamento de erro:

- Não permitir que transferências sejam feitas com saldo insuficiente.
- Não permitir o saldo de quaisquer umas das contas ser negativo

(Obs: Por hora, não é necessário preocupar com senhas para proteger as contas, o objetivo aqui é utilizar tudo que foi aprendido e forçar o discente a ter uma visão melhor de criação de um programa.)

(Segunda Obs.: como já foi requisitado a exibição do saldo de ambas as contas, não é necessário registrar as transferências ou saques e depósitos feitos, conhecido comumente como extrato bancário.)