

Prova 1

- Escreva as respostas com caneta.
- Você pode criar funções auxiliares, se desejar. Contudo, não considere a pré-existência de funções prontas e não utilize funções da biblioteca além de `printf()` e `scanf()`.
- Atribua de 1,0 à 3,0 pontos para cada questão, não ultrapassando o total de 10,0 pontos.

1) () Escreva um programa que lê três números, A, B e C. Depois, ele deve imprimir: o maior entre eles, bem como, a paridade e o sinal de cada número. Observe o exemplo abaixo:

Entre com 3 números inteiros: 5 9 -2

Maior: 9

5 é positivo e ímpar

9 é positivo e ímpar

-2 é negativo e par

2) () Escreva um programa que faz a leitura de vários números inteiros, até que ZERO seja digitado.

Encerrada a entrada (com o número zero), o programa deve exibir um relatório, contendo:

- a) A soma dos positivos
- b) A soma dos negativos
- c) A diferença entre o cubo dos positivos e o quadrado dos negativos ($\text{positivos}^3 - \text{negativos}^2$)
- d) A quantidade de números informados (exceto 0)
- e) A quantidade de números ímpares informados

3) () Em uma agência bancária, as contas são identificadas por números de até 7 dígitos, seguidos de dois dígitos verificadores (veja processo abaixo). Dado um número de conta n, exiba o número de conta completo. OBS: para imprimir a conta, use `printf("%07d-%02d", conta, digito);`

- a) Seja $n = 190450$ o número da conta (considere que n será um número positivo).
- b) Utilize divisão inteira (/) e módulo (%) para obter os dígitos, até que o quociente seja 0
- c) Multiplique os dígitos de n **diferentes de zero** para obter $t = 1*9*4*5 = 180$;
- d) Utilize o resto da divisão de t por 100 para obter o verificador $180 \% 100 = 80$.
- e) Número de conta completo: 0190450-80

4) () Escreva um programa que lê o número de **linhas** para desenhar um triângulo de multiplicação.

Cada valor é dado por linha x coluna, começando sempre por 1. OBS: para imprimir o número alinhado à direita, preenchido com zeros, utilize `printf("%02d ", num)`.

Exemplo: linhas = 7

01	02	03	04	05	06	07
02	04	06	08	10	12	
03	06	09	12	15		
04	08	12	16			
05	10	15				
06	12					
07						