

### Observações

- Os programas devem ser elaborados em *Portugol* ou em *linguagem C*, ou em *Python*.
- Utilize técnicas para a elaboração de algoritmos estruturados e legíveis: utilize estruturas de controle de fluxo adequadas, inclua comentários nos algoritmos aonde for necessário, use indentação, escreva apenas um comando por linha, defina variáveis com identificadores claros, etc.

Ao final, compactar o arquivos (.c ou .cpp) e enviar no AVA.

1. Crie um programa para ler 3 números do teclado e informar se estes números podem formar um triângulo. Se puder formar um triângulo, mostrar se formam um triângulo equilátero, isósceles ou escaleno.  
*Obs: Para formar um triângulo um número não pode ser maior que a soma dos outros dois.*
2. Crie um programa para ler um número inteiro do teclado e mostrar a soma dos números contidos entre 0 (zero) e esse número, a soma somente dos números pares e a soma somente dos números ímpares.
3. Crie um programa para ler um número com quatro dígitos e escrever em separado o dígito que representa a milhar, a centena, a dezena e a unidade.
4. Crie um programa para ler números do teclado, o programa deve ir lendo números do teclado e finalizar quando for digitado 0 ou um número negativo. Depois deve mostrar a quantidade de números lidos, o maior número, o menor, a soma e a média.
5. Uma cidade tem 2 candidatos à prefeitura. Crie um programa para ler o total de eleitores desse município, depois ler o número de votos do candidato 1, do candidato 2 e o número de votos nulos. O número de votos brancos será o total de eleitores menos os votos nos candidatos e os nulos. Ao final, mostrar o percentual de votos de cada candidato, o percentual de votos nulos e de votos em branco. Todos esses percentuais devem ser em relação ao total de eleitores. Mostrar também qual o candidato vencedor ou se houve empate.
6. Crie um programa para ler uma palavra do teclado e informar quantas vogais existem nessa palavra.
7. Escreva um programa para conter um vetor com 20 posições, preencher cada posição com números aleatórios entre 1 e 100 e deve conter um menu com as seguintes opções:
  1. Digitar um número
  2. Mostrar só os números pares
  3. Mostrar só os números ímpares
  4. Mostrar só os números das posições pares do vetor
  5. Mostrar só os números das posições ímpares do vetor
  9. Finalizar

Obs: criar o vetor global e em cada vez que mostrar o que foi pedido pelo item de menu digitado, mostrar também o conteúdo de todas as posições do menu.

8. Crie um programa que receba um número inteiro positivo  $n$  e apresente um quadro de números como o que se segue abaixo. Este é o caso de  $n = 5$  (pode-se limitar o valor em 10).

Exemplo ( $n=5$ ):

1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8
5	6	7	8	9

9. Escreva um programa para ler uma palavra do teclado e mostrar se é um palíndromo.

*Obs: palíndromo, ou capícuo, ou cabeça-cauda é uma palavra que é igual tanto da esquerda pra direita como da direita pra esquerda, exemplos: OVO, RADAR, ANA, ARARA, LUZAZUL, OTTO, SUBINOONIBUS, AMOR A ROMA, OTTO AMA ANA E ANA AMA OTTO, SOCORRAMMARROCOS, SOCORRAMMESUBINOONIBUSEMMARROCOS.*

Dica: para saber se uma palavra é palíndromo, comece marcando uma variável de controle (*flag por exemplo*) para TRUE e depois percorra a palavra do começo ao fim e do fim pro começo, isto pode ser feito com um laço de repetição só e até a metade da palavra. A cada passo do laço você compara a primeira letra com a última, se for igual não faz nada, se for diferente então você encontrou uma diferença, então você marca a sua variável *flag* para FALSE.

De novo, neste laço, compara a primeira letra com a última, depois a segunda com a penúltima, a terceira com a antepenúltima... e por aí vai, até chegar no meio da palavra, se em alguma comparação encontrar letras diferentes então você marca a sua variável *flag* para FALSE.

Ao final você verifica se a sua variável *flag* é TRUE, então a palavra é palíndromo, senão não é.

