

Orientações: Nosso cérebro não se encontra na ponta dos nossos dedos, dito isso e lembrando de nossa 1ª e 2ª aulas, sugiro:

Ao fazer um dos nossos exercícios de programação, primeiro faça uma análise prévia:

- a) O que o programa pede, qual o objetivo?
- b) identifique as variáveis de entrada e seus tipos de dados, dê nomes coerentes a elas.
- c) qual o processamento? o que será feito com as variáveis de entrada, cálculos, etc.
- d) identifique a(s) variáveis a serem apresentadas na saída, qual o tipo? Qual a forma de apresentação?

- Utilize o *template* sugerido, que está a seguir:

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
void main(void)
{
    setlocale(LC_ALL, "Portuguese");
}
```

Após essa breve análise, só então passe para a codificação.

Como última recomendação, sugiro recorrerem aos livros sugeridos, aos slides, a min, ou ainda aos colegas, mas não consultem internet ou Chat GPT. Nessas duas últimas opções certamente encontrarão as respostas, mas esses programas iniciais, servem para que desenvolvam um raciocínio objetivo e preciso no que se refere à programação e sedimentem a sintaxe da linguagem e desenvolvam sua capacidade de depuração (“debugar”) os programas

Os exercícios constantes da próxima folha se referem somente a execução em sequência, assim como fizemos em sala de aula. Para dar nome aos seus arquivos façam da seguinte forma:

Legenda:

ex_nn_6_vet_t#_rrrrrrr_nome.c, onde:

nn – número do exercício

- a, para a turma a e assim sucessivamente.

rrrrrrr – RA do aluno, somente números.

nome – nome do aluno_primeirosobrenome.

Tudo em minúsculo, não serão aceitos exercícios realizados de outra forma!

Exemplo: Aluno Tony Stark, da turma C que vai dar nome ao exercício 1

ex_01_6_vet_ta_12345678_tony_stark.c

Após realizar todos os exercícios da folha, os compacte em formato zip e enviem para o meu e-mail (miguel.archanjo@ceub.edu.br), a data de entrega será informada em mensagem específica para a turma, o arquivo compactado deverá ter o seguinte nome:

t#_vet_rrrrrrr_nome.zip , seguindo a mesma legenda acima, no caso do aluno Tony Stark acima, seria: ta_vet_12345678_tony_stark.zip

Exercícios de Vetor

1. Leia quatro valores digitados pelo usuário, armazene-os num vetor e no final mostre esses valores que estão armazenados no vetor.
2. Refaça o programa anterior, mostrando os valores armazenados no vetor na ordem inversa da entrada.
3. Construa um programa que calcule a média aritmética de uma classe com dez alunos, onde cada aluno teve uma avaliação (use vetor). Gere um relatório, tela de saída, com o número e a nota dos alunos e no final mostre também a média da turma.
4. Refaça o programa anterior para mostrar também a quantidade de notas acima da média da turma.
5. Elabore um programa que leia, some e imprima o resultado da soma posicional, entre dois vetores inteiros de 10 posições. Use um terceiro vetor para armazenar o resultado.
6. Construa um programa que preencha um vetor de 100 elementos inteiros, colocando "1" na posição (= índice do vetor) correspondente a um número par e "0" na posição (= índice do vetor) correspondente a um número ímpar.
7. Dados dois vetores de inteiros, "a" com dez elementos e "b" com vinte elementos, escreva um programa que faça a união destes dois vetores armazenando todos os elementos num terceiro vetor. Leia os dados dos vetores "a" e "b" e no final mostre o vetor união.
8. Selecione os elementos comuns aos dois vetores acima, armazenando-os num terceiro vetor. No final mostre os valores selecionados, ou seja, os elementos comuns.
9. Elabore um programa que, dados dois vetores inteiros de 20 posições, efetue as respectivas operações indicadas por um outro vetor de 20 posições de caracteres também fornecido pelo usuário, contendo as quatro operações aritméticas em qualquer combinação, armazenando os resultados num quarto vetor.
10. Leia uma palavra e mostre-a na ordem inversa. Utilize a função comprimento da string.
11. Elabore um programa para limitar a quantidade máxima de caracteres que poderemos digitar para uma entrada alfanumérica, ou seja, variável string. O usuário fornecerá uma palavra ou uma sentença que será finalizada quando ele digitar a tecla < enter > ou quando chegar ao limite máximo de sessenta caracteres digitados. No final mostre a quantidade de caracteres efetivamente digitados