

 INSTITUTO FEDERAL Paraíba	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA – CAMPUS CAMPINA GRANDE		
	CURSO:		
	PERÍODO		TURMA:
	DISCIPLINA:		
	PROFESSOR:	CESAR VASCONCELOS	SEMESTRE LETIVO: 2023.1

GitHub: <https://github.com/cesarvasconcelos/cplus-poo>

Usando C++, implemente os algoritmos (utilize **vector**, **structs**, **módulos** e **funções** onde achar necessário):

1. Escreva um programa para ler uma quantidade de números inteiros desconhecida como entrada e possa calcular a *soma* destes números. Armazene os números de entrada em um vector. O programa deve imprimir a lista de entrada e a soma final calculada. Para este algoritmo, a leitura do valor zero (0) indicará o final da leitura de entrada de dados. Use funções e módulos para organizar o programa.
2. Reescreva a questão 1 anterior pesquisando na *Web* ou em documentação se, no C++, existe alguma função utilitária já pronta da linguagem capaz de somar os números de um vector.
3. Reescreva a questão 1 anterior de forma que o programa deve agora ser capaz de informar ao usuário a média dos números da lista (vector) e ainda montar e imprimir uma sub-lista contendo apenas os números que estiverem acima da média calculada. Verifique se, no C++, existe alguma função utilitária da linguagem capaz de calcular a média dos números de uma lista como um vector.
4. Assuma que seu programa possui a seguinte lista de nomes de clientes da empresa:

```
clientes = ['caio', 'wellington', 'pedro', 'bruno', 'ana', 'caio', 'bruno', 'david', 'pedro']
```

- Na documentação de C++ na Web, pesquise qual é a função responsável por comparar duas Strings.
 - Escreva uma função que seja capaz de verificar se um dado nome passado como entrada se encontra na lista/vector e, se existir, informar o número de ocorrências.
 - Escreva uma função que seja capaz de receber um nome passado como entrada e remover todas as ocorrências deste nome na lista
 - Escreva uma função capaz de receber um nome passado como entrada e possa retornar a posição da primeira ocorrência de um dado nome na lista. Se o nome não existir, deve-se retornar o valor -1 sinalizando posição inválida não encontrada
 - Escreva uma função capaz de receber um nome passado como entrada e possa retornar a posição da última ocorrência de um dado nome na lista. Se o nome não existir, deve-se retornar o valor -1 sinalizando posição inválida não encontrada
5. Reescreva a questão anterior usando funções em módulos, e agora modelando o nome do cliente e ainda sua idade usando os fundamentos de **struct**; Agora, o vector irá guardar structs em vez de strings. Identifique funções e as agrupe em módulos para operar sobre essa struct corretamente e reduzir a lógica do código dentro do main()
 6. Uma estação meteorológica deseja construir um algoritmo que leia a temperatura média (exemplo 27.5 Celsius) de cada mês do ano. Ao final, o programa deve exibir na tela as seguintes informações:
 - a. Todas as temperaturas médias que foram passadas como entrada para o programa;
 - i. Obs.: Armazene as temperaturas médias mensais em um vector em C++.

- b. Temperatura média anual;
 - c. Quantidade de temperaturas mensais superiores a temperatura média anual;
 - d. Os nomes dos meses com temperaturas superiores a temperatura média anual;
 - e. Quantidade de temperaturas mensais inferiores a temperatura média anual;
 - f. Quantidade de temperaturas mensais iguais a temperatura média anual (se houver).
 - g. use funções e módulos nesta questão, para reduzir a lógica de código no main()
7. (Para profissionais!) Reescreva a questão anterior usando funções em módulos, e ainda modelando a estação meteorológica, e suas temperaturas mensais usando os fundamentos de **struct**; identifique funções e as agrupe em módulos para operar sobre essa struct corretamente e reduzir a lógica do código dentro do main().
8. Faça um programa para uma escola que, dadas três notas obrigatórias de um aluno e seu nome completo, exiba, no final, o nome do estudante, notas, a média final e o seu conceito, observando que:
- a. Todos os dados de entrada do aluno devem ser armazenados **em uma lista (vector)**;
 - b. As 3 notas devem ser guardadas **após o nome do aluno**;
 - c. O nome lido como entrada deve ser guardado **na primeira posição** da lista;
 - d. a média final é calculada a partir da média aritmética simples das 3 notas e deve ser guardada **após a terceira nota do aluno** na lista
 - e. o conceito deve ser guardado **na última posição** da lista

Exemplo: `vec_aluno = {'caio', 80, 40 , 90, 70, 'B' } # [nome, nota1, nota2, nota3, média, conceito]`

- f. o conceito é determinado a partir da tabela a seguir:

MÉDIA FINAL	CONCEITO
≥ 80	A
$\geq 50 \text{ e } < 80$	B
< 50	C

- g. A mensagem final exibida do sistema deve ter o seguinte formato (substitua os espaços entre <> pelos respectivos valores passados como entrada para o sistema):

“<Fulano de tal> **obteve conceito** <X>”

“**As 3 notas fornecidas como entrada foram:** <sub-lista com 3 notas> **com Média final:** <Y>”

9. (Para profissionais!) Reescreva a questão anterior usando funções em módulos, e ainda modelando o aluno, suas notas e o conceito usando os fundamentos de **struct**; identifique funções e as agrupe em módulos para operar sobre essa struct aluno corretamente e reduzir a lógica do programa do main().
10. Faça um programa usando funções/módulos que, dada uma figura geométrica (circunferência, triângulo ou retângulo), calcule e exiba a área e o perímetro da figura. O programa deve, primeiro, perguntar em um menu ao usuário qual o tipo da figura:
- (1) circunferência
 - (2) triângulo
 - (3) retângulo

Dependendo do tipo de figura que o usuário escolher, o programa deve fazer a leitura de dados específica da figura. Por exemplo, para a figura (1), deve-se ler o raio da circunferência; Para a figura (2), o programa deve ler o tamanho de cada um dos lados do triângulo; E, finalmente, para a figura (3), o programa deve ler do usuário a base e altura do retângulo.

Observação: pesquise sobre as fórmulas para determinar a área e o perímetro de cada figura.

A leitura dos dados específicos das figuras, o cálculo de área e perímetro, bem como a impressão individualizada do processamento de cada figura devem ser feitos por meio de funções e módulos

11. Uma empresa de mineração faz, periodicamente, um estudo estatístico sobre a produção de urânio em suas minas. A empresa possui duas minas (uma situada ao norte e outra no sul). Escreva um algoritmo que leia a produção de urânio em toneladas (exemplo: 12.93) de cada dia da semana de uma mina e guarde todos estes valores em uma lista (1 para cada mina). O algoritmo, no final, deve ser capaz de imprimir:
 - a. A produção diária de cada uma das minas
 - b. A produção média semanal de urânio da mina do norte e da mina do sul;
 - c. Qual mina possui a maior produção semanal média de urânio em toneladas.
12. (Para profissionais!) Reescreva a questão anterior usando funções em módulos, e ainda modelando a mina, sua localização e sua produção semanal usando os fundamentos de **struct**; identifique funções e as agrupe em módulos para operar sobre essa struct corretamente e reduzir a lógica do main().
13. Escreva um programa que leia um conjunto de números inteiros de tamanho desconhecido e os **armazene em uma lista (vector)**. O programa deve ser capaz de informar ao usuário quantos números pares existem na lista. Um número é considerado par quando o resto da divisão inteira por 2 (dois) é igual a 0 (zero), ou seja: $(num \% 2) == 0$ Para este algoritmo, a leitura do valor zero (0) indicará o final da leitura de entrada de dados
14. Escreva um algoritmo que, mediante uma lista/vector de números inteiros não ordenada, possa informar a posição (o *index*) do maior elemento dentro da lista.
15. Faça um programa que leia 5 valores correspondentes a notas (0 a 100) de alunos e os armazene em uma lista (vector). Após esta entrada de dados, implemente em módulo:
 - uma função que receba a lista e retorne a média das notas;
 - uma função que receba a lista e retorne a quantidade de notas abaixo de 70;
 - uma função que receba a lista e retorne a quantidade de valores acima da média.

Obs.: faça a chamada de cada função, apresentando o resultado do processamento na tela

16. Escreva um algoritmo para ler uma quantidade de números inteiros desconhecida como entrada e possa armazená-los em uma lista. O programa deve imprimir a lista de entradas de forma ordenada, e ainda remover todos os números negativos antes da impressão. Para este algoritmo, a leitura do valor zero (0) indicará o final da entrada de dados.