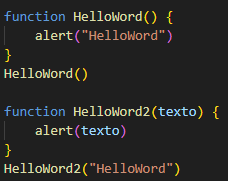
**Dimitri Diciocio**

**Resolução das atividades de JavaScript**

Considerações Iniciais: Durante a explicação das atividades, ative-me a explicar linha por linha do codigo, entretanto, linhas comuns serão comentadas a seguir, com o objetivo de diminuir a necessidade de explicações constantes e repetitivas. Todos os comentários essenciais das questões tem links diretos a bibliografia com a explicação, junto de uma explicação parafraseada dos links.

Funções: [LINK](https://www.w3schools.com/js/js_functions.asp) ou w3schools.com/js/js\_functions.asp

Conforme o exemplo a seguir, as funções tem o objetivo de guardar linhas de codigo, que serão executadas ao chamar uma função, tornando o codigo mais organizado e criando diversas outras funcionalidades, como mudar a resposta do codigo dependendo da informação, como no exemplo 2. As funções também tem um papel importantissímo na programação, quase todo o básico aprendido, como alertar na tela(Será visto na proxima consideração) é uma função, propria do JavaScript ou importada junto de sua biblioteca, que faz, por baixo dos panos, uma sequência de codigos que leva ao resultado. Tudo que utilizar este padrão a seguir, é uma função: texto( ), como alert( ), console.log( ). A seguir um codigo de exemplo, alertando hello word.

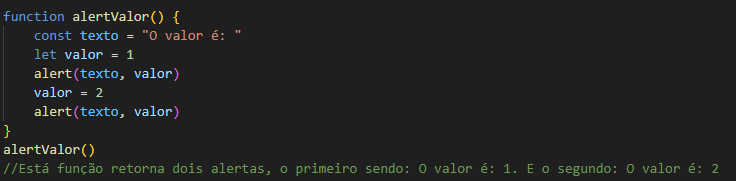


Alertas: (Visto em aula)

alert( ) é uma função, que retorna um pop-up com uma informação, escrita em uma variavel ou diretamente entre “aspas”. Como visto no Hello Word acima.

Let e Const: [LINK](https://stackoverflow.com/questions/76010295/when-should-you-use-var-let-or-const-in-javascript-code) ou stackoverflow.com/questions/76010295/when-should-you-use-var-let-or-const-in-javascript-code

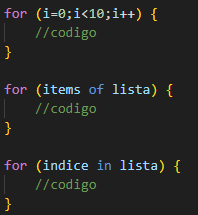
Atualmente, as declarações, let e const são mais utilizadas. Let é uma declaração que cria uma variavel em seu escopo, como um for e um if, sem ser utilizada fora de seu escopo, semelhante ao var, porém mais recente nos padrões do ECMAScript. Const vem para declarar constantes, coisas que não podem ser mudadas, portanto, servem para valores que não são variáveis, estáticos. No exemplo a seguir temos os dois sendo utilizados.



Como é possível ver, const é um valor que não muda, imutável, e let um valor que pode ser mudado. Outro ponto importante, tanto let quanto const só vão ser utilizados na sua criação, a partir disso, não é preciso declarar const ou let para o valor, pois ele já representa um const ou um let.

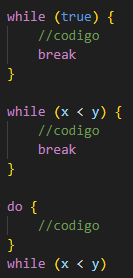
For loop: [LINK](https://www.w3schools.com/js/js_loop_for.asp) ou w3schools.com/js/js\_loop\_for.asp

Existem 3 formas de declarar um for loop, for in, for of e for. Começando com a definição de for, for é um bloco de repetição, que repete um bloco de codigo um numero definido de vezes. A declaração padrão do for é for (variavel = quantidade; varivel <= quantidade2; variavel++), que significa, varivel recebe quantidade, enquanto variavel ser menor ou igual a quantidade2, variavel recebe variavel + 1. O que faz com que seja possível repetir o bloco de codigo dentro um numero de vezes. Já o for in, usado para fazer um bloco de repetição que tem um tamanho definido e uma variavel definidas a partir de uma variavel, exemplo, for (variavel in lista), isso repe variavel, começando em 0, até o ultimo indice da lista, ou seja, a posição de seu ultimo elemento. For of é semelhante, declarado por for (variavel of lista), variavel recebe o valor referente ao indice da lista, não apenas a posição dele.



While loop: [LINK](https://www.w3schools.com/js/js_loop_while.asp) ou w3schools.com/js/js\_loop\_while.asp

Um while loop é um bloco de repetição, semelhante ao for, mas que repete até que determinada condição seja atingida, sendo declarado da seguinte forma while (variavel < quantidade), que repete até que a variavel, dentro do loop, atinja a condição, ou while (true), que repete até que um break aconteça dentro do bloco de repetição. Também existe o do while, escrito da seguinte forma, do {codigo} while (condição), semelhante porém com outra forma de escrita.

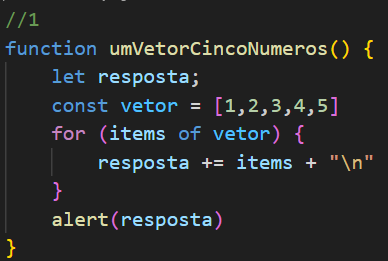


Visto esses pontos, a seguir as atividades explicadas detalhadamente:

1. Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.

[LINK](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array) ou [developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\_Objects/Array](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array)

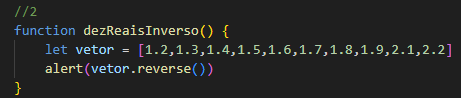
Começamos com a pergunta, o que é um vetor ? Simples, um vetor é um array, também chamado de lista, determinado em JavaScript com o uso dos colchetes [ ]. Portanto, para a resolução, foi criado uma função(Explicada nas Considerações Iniciais) com um nome explicito: umVetorCincoNumeros, a criação, com const, pela não necessidade de mudança da lista nessa atividade, um vetor contendo 5 numeros inteiros, com o nome “vetor”, dando a característica ao vetor de um array de inteiros. Para mostra-los, faremos um for of loop para conseguir os items de dentro do array, colocando-os em uma variável resposta, em seguida, utiliza-se do alert, passando resposta como argumento da função, exibindo os items do vetor na tela.



2 - Faça um Programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.

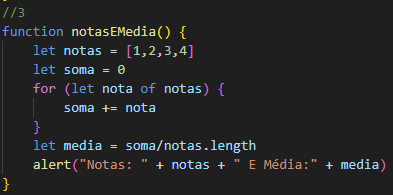
[LINK](https://awari.com.br/guia-completo-sobre-classes-em-javascript-tudo-o-que-voce-precisa-saber/#:~:text=Em%20JavaScript%2C%20os%20m%C3%A9todos%20s%C3%A3o,dentro%20da%20defini%C3%A7%C3%A3o%20da%20classe.&text=Al%C3%A9m%20dos%20m%C3%A9todos%2C%20tamb%C3%A9m%20) ou https://awari.com.br/guia-completo-sobre-classes-em-javascript-tudo-o-que-voce-precisa-saber/#:~:text=Em%20JavaScript%2C%20os%20m%C3%A9todos%20s%C3%A3o,dentro%20da%20defini%C3%A7%C3%A3o%20da%20classe.&text=Al%C3%A9m%20dos%20m%C3%A9todos%2C%20tamb%C3%A9m%20temos%20as%20propriedades%20em%20uma%20classe%20JavaScript.

A diferença principal das duas atividades, mostrar na ordem inversa. A segunda é o que são numeros reais, começarei da mais facil. Numeros reais são numeros que pertencem ao conjunto dos reias, sendo eles, numeros com ponto flutuante, como 2.2(representado na programação através de um ponto, ao invés de uma virgula, por ser a forma utilizada em inglês, além de trazer o nome ao tipo na programação, float, que significa ponto flutuante, pois o valor não é inteiro, mas com um ponto, que indica um numero entre inteiros, contendo um ponto(virguça) flutuando entre os numeros.), numeros inteiros, como -1, e naturais como 1. Voltando a diferença principal, que é bem simples, entretanto, por ser nova, pode causar confusão ao tentar resolver, seu objetivo é inverter a ordem de uma lista, o que é simples, por existir um método pertencente aos arrays, chamado reverse, que inverte o array. Como mostra na resolução da atividade a seguir, com o vetor.reverse( ). Surge dai outra duvida, por que é escrito dessa forma, nomeDaVariavel.Metodo( ), a resposta é que reverse é um metodo que pertence ao array, como dito anteriormente, portanto, o array se trata de um objeto, que possui uma função propria, chamada reverse, que pode ser chama apenas para um array, a isso se da o nome Método, nesse sentido, o método do vetor, que é um array de inteiros, chamado reverse, inverte a ordem desse vetor, tornando possível fazer essa atividade sem um for in loop de vetor passando para outro array em ordem inversa para então trocar o valor do vetor pelo outro invertido.



3 - Faça um Programa que leia 4 notas, mostre as notas e a média na tela.

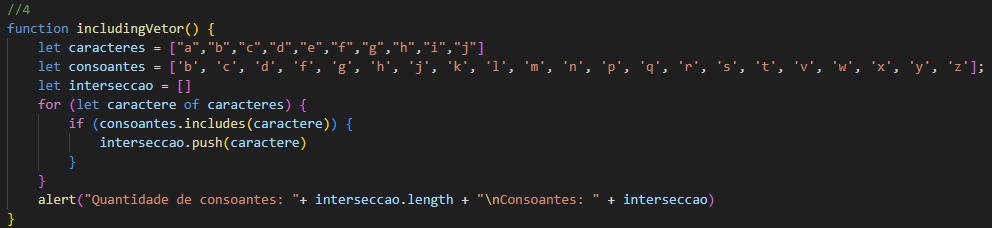
A primeira vista, parece complexo, mas é semelhante aos outros. O que pode causar confusão é a forma como faremos para calcular a media. Daremos inicio declarando o array com as quatro notas e a variavel soma como 0, que será usada mais para frente, em seguida, faremos um for of com o array, que acessará todas as suas informações uma por uma, assim, declaramos que soma recebe soma mais informação, até o fim do array, conseguindo assim encontrar a soma dos valores do array, no fim do loop, criaremos a variavel media, que recebe na sua criação, soma dividido por array.length, que é um metodo do array que conta quantos elementos possui dentro do array, resultando na media. Por fim um alert para mostrar na tela, concatenando a string “Notas: “ com o array notas e concatenando também a string “E Média: ” com media.



4 - Faça um Programa que leia um vetor de 10 caracteres, e diga quantas consoantes foram lidas. Imprima as consoantes.

[LINK](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/includes) ou https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\_Objects/Array/includes

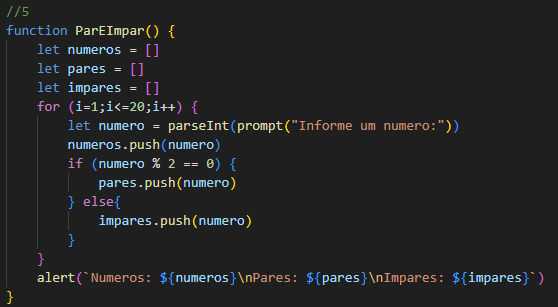
Nesta questão, é necessário criar 3 vetores, um vazio, um com os 10 caracteres, e outro com todas as consoantes. Para então, com um for of, seja possível comparar as consoantes dentro da lista de 10 caracteres, com as consoantes dentro da lista de consoantes. Para isso, iniciamos o for of, e dentro dele, faremos uma checagem com if se o item dentro da lista correspondente no for, está incluso na lista de consoantes, para isso, utilizaremos o método includes, que checa se dentro do array consoantes, existe a variavel passada para o metodo, que é a informação, se existir, armazenaremos, com o método push do array, que adiciona na ultima posição da lista, a variavel que passamos, que é a informação, adicionando então a consoante no vetor vazio, com isso, ao final do for, teremos uma lista com todas as consoantes da lista, então, passamos um alert informando a quantidade de elementos do array novo, com isso, passamos a quantidade de consoantes, e então, passamos o array em si, imprimindo as consoantes pertencentes ao vetor de 10 caracteres.



5 - Faça um Programa que leia 20 números inteiros e armazene-os num vetor. Armazene os números pares no vetor PAR e os números IMPARES no vetor impar. Imprima os três vetores.

[LINK](https://www.javascriptprogressivo.net/2018/07/Como-Receber-Dados-Usuario-Metodo-prompt.html) ou https://www.javascriptprogressivo.net/2018/07/Como-Receber-Dados-Usuario-Metodo-prompt.html

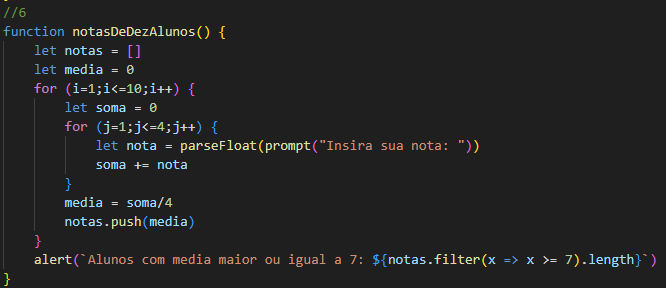
Essa atividade não tem tantas diferenças, apenas na forma como faremos, para garantir um novo aprendizado, o recebimento das informações, utilizando a função prompt e parseInt, que fazem, respectivamente, o pedido de informações e a transformação da informação em inteiro. Para fazer essa atividade, faremos um for loop, de 20 numeros, pedindo 20 vezes, com parseInt(prompt( )) numeros inteiros, em seguida, guardamos essa informação dentro de um array com push( ), faremos a checagem se o numero é impar ou par, com if, utilizando % para encontrar o resto da divisão, descobrindo se o numero é par ou impar, e então guardando no seu respectivo array, ao final, um alert passando os três arrays.



1. Faça um Programa que peça as quatro notas de 10 alunos, calcule e armazene num vetor a média de cada aluno, imprima o número de alunos com média maior ou igual a 7.0.

[LINK](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter) ou https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/filter

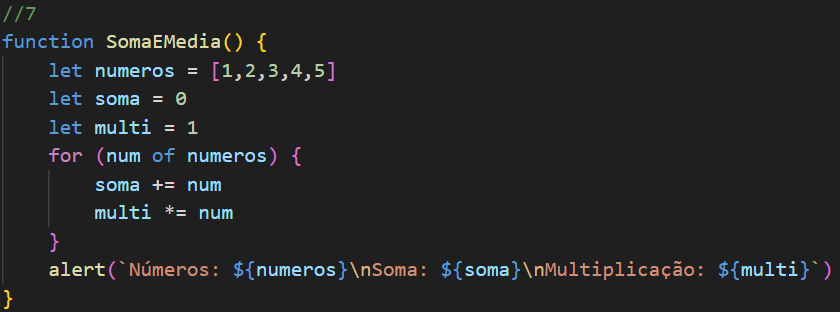
Para essa atividade, precisaremos imprimir(alertar) o número de alunos com media maior ou igual a 7, portanto, a diferença entre essa e a de cima é apenas o cálculo da média, e encontrar o número de alunos, portanto, vou focar apenas nisso. Para fazer o cálculo da média de cada aluno, faremos dois for loops, um dentro do outro, para repetir 10 vezes um loop de 4 vezes. Assim pedindo 4 notas de 10 alunos, ao final do loop aninhado, que faz o método de soma das atividades anteriores(atividade 3), dividimos por 4, encontrando a média das notas e salvando com push( ) em uma lista de notas. Passando para a forma de imprimir o número de alunos, usamos o método filter, que percorre nosso array de medias e checa cada nota, se a nota avaliada estiver de acordo com o check, ele a salva em um array, assim, passamos no metodo filter o metodo length, checando a quantidade de elementos dentro do array criado pelo metodo filter, resultando no numero de alunos com a media maior que 7.



7- Faça um Programa que leia um vetor de 5 números inteiros, mostre a soma, a multiplicação e os números.

[LINK](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Template_literals) ou https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Template\_literals

Essa atividade, por ser muito semelhante a anterior, vou me abster de explica-la. Focando apenas no novo, a utilização dos acentos graves, para a escrita da resposta( também foi utilizado na atividade passada, mas deixei para explicar nessa devido a semelhança). A utilização dos acentos graves na criação de strings, permite informar variáveis dentro da string, utilizando ${variavel}, com ele, também é possível formatar com espaços e quebras de linha de forma direta, sem precisar utilizar necessariamente o \n como quebra.



8- Faça um Programa que peça a idade e a altura de 5 pessoas, armazene cada informação no seu respectivo vetor. Imprima a idade e a altura na ordem inversa a ordem lida.

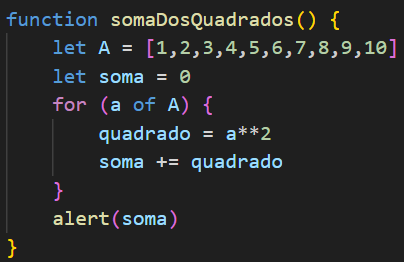
Criei a lista de idades e alturas, fiz um for loop para encontrar as idades, coloquei a idade e a altura nas suas respectivas listas e dei um alert com as idades com o método reverse e as alturas também.



9- Faça um Programa que leia um vetor A com 10 números inteiros, calcule e mostre a soma dos quadrados dos elementos do vetor.

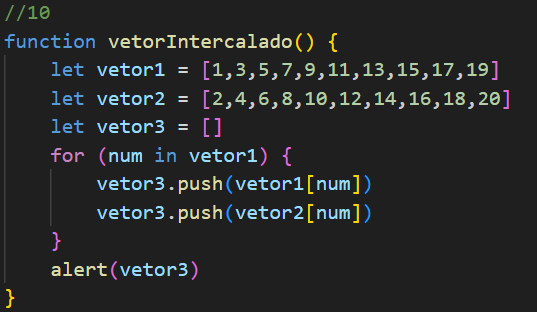
[LINK](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/Exponentiation) ou https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Operators/Exponentiation

Para essa atividade, precisamos fazer a soma dos quadrados, portanto, criamos uma lista com os números inteiros, uma variável para a soma sendo 0, um for of do vetor A ( vetor contendo os números), criamos a variável quadrado, que é igual ao numero \*\* 2, o símbolo \*\* significa exponenciação, conforme mostra o link acima, após isso, colocamos soma += quadrado, para somar os quadrados, assim, alertamos soma e temos o resultado.



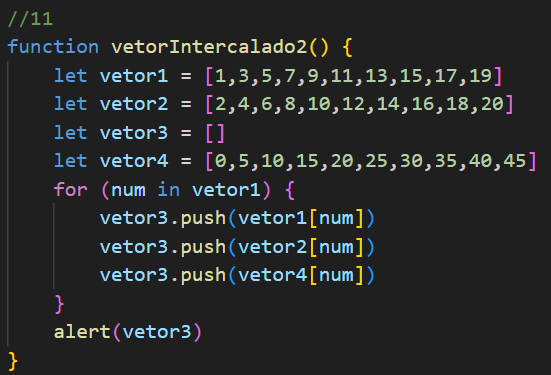
10- Faça um Programa que leia dois vetores com 10 elementos cada. Gere um terceiro vetor de 20 elementos, cujos valores deverão ser compostos pelos elementos intercalados dos dois outros vetores.

Para essa atividade, criaremos 3 vetores, dois contendo números inteiros, e um vazio, depois um for in do vetor 1, em que colocamos no vetor vazio, o vetor 1 e 2 na posição num criada no for, ao final, alertamos o vetor 3.



11- Altere o programa anterior, intercalando 3 vetores de 10 elementos cada.

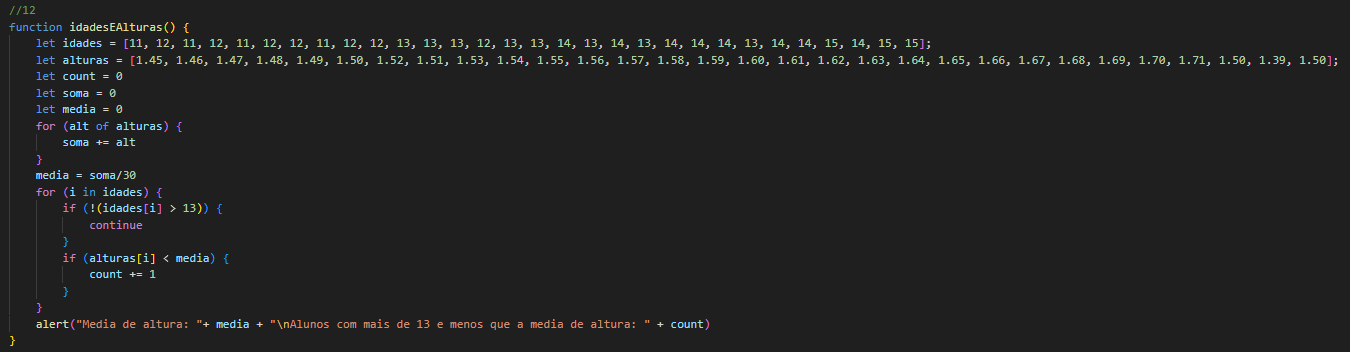
Repita a atividade anterior, colocando um vetor a mais, assim, resolvendo a questão de forma simples.



12- Foram anotadas as idades e alturas de 30 alunos. Faça um Programa que determine quantos alunos com mais de 13 anos possuem altura inferior à média de altura desses alunos.

[LINK](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/continue) ou https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/continue

Crie um array contendo números inteiros para idade, um com alturas, três variáveis, uma chamada count, outra soma e outra media, faça uma for of de alturas, somando as alturas dentro do for, encontre a media com soma/30, crie um for de idades para checar idades maiores que 13, se idade ser menor que 13, ele passa para o próximo do loop com o continue, como no link acima. Se não, ele checa a altura. Se ela for menor que media, adicionamos um em count, ao fim, alertamos media e count.



13- Faça um programa que receba a temperatura média de cada mês do ano e armazene-as em uma lista. Após isto, calcule a média anual das temperaturas e mostre todas as temperaturas acima da média anual, e em que mês elas ocorreram (mostrar o mês por extenso: 1 – Janeiro, 2 – Fevereiro, . . . ).

Para essa atividade, criamos um array vazio para armazenar as temperaturas, uma variável de soma, uma lista vazia de acima da media, e um array com os meses escritos por extenso. Crie um for de 10 numeros. Pedindo a temperatura media do mês, guardando no array de temperaturas e somamos a temperatura pedida na variável soma. Ao final do for, fazemos a media, com soma/12, um for in com as temperaturas checando com if se a temperatura está acima da media. Se ela estiver, colocamos na lista acima de acima da media o nome por extenso do mês, e a temperatura. Para passar no alert o acima da media forma apenas um alert.



14- Utilizando listas faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são:

"Telefonou para a vítima?"

"Esteve no local do crime?"

"Mora perto da vítima?"

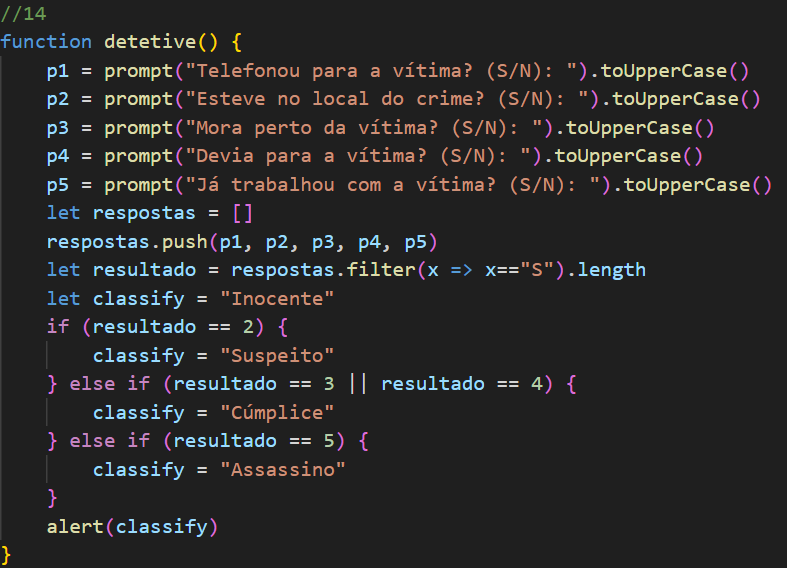
"Devia para a vítima?"

"Já trabalhou com a vítima?"

O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice".

[LINK](https://www.w3schools.com/jsref/jsref_touppercase.asp) ou https://www.w3schools.com/jsref/jsref\_touppercase.asp

Para começar, criei 5 variáveis com prompts de pergunta, ao final da variável, utilizo o metodo .toUpperCase( ), que coloca a resposta como caixa alta(maiúsculo), criei outro array vazio chamado respostas, onde coloquei todas as respostas com um push, criei a variável resultado, com um filter de S, criei uma variável classify, com a string Inocente, fiz um if check do resultado, se for igual a 2, classify é igual a suspeito, assim para o resto das classificações de participação do crime.



15- Faça um programa que leia um número indeterminado de valores, correspondentes a notas, encerrando a entrada de dados quando for informado um valor igual a -1 (que não deve ser armazenado). Após esta entrada de dados, faça:

Mostre a quantidade de valores que foram lidos;

Exiba todos os valores na ordem em que foram informados, um ao lado do outro;

Exiba todos os valores na ordem inversa à que foram informados, um abaixo do outro;

Calcule e mostre a soma dos valores;

Calcule e mostre a média dos valores;

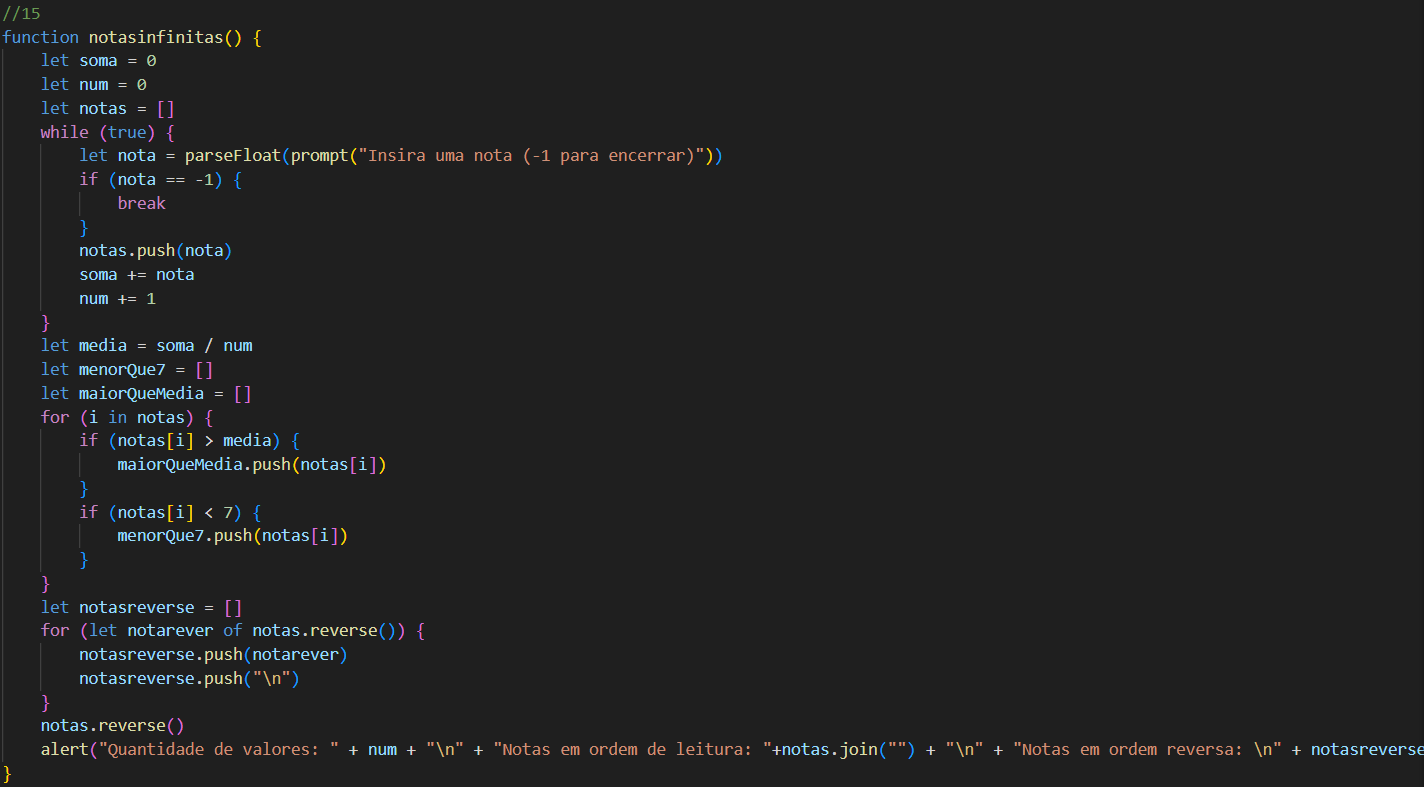
Calcule e mostre a quantidade de valores acima da média calculada;

Calcule e mostre a quantidade de valores abaixo de sete;

Encerre o programa com uma mensagem;

[LINK](https://www.w3schools.com/jsref/jsref_join.asp) ou <https://www.w3schools.com/jsref/jsref_join.asp>

Para essa atividade, criamos 2 variaveis, um chamado soma, e outro num, ambos valendo 0, um array de notas. Criamos um while true para ler com prompt notas e armazenar na variável nota, se nota for igual a -1, ele para o while, utilizando um if e um break para isso. Também guardamos a nota no array notas, somamos em soma a nota, e colocamos num += 1 para contar o numero de repetições. Ao final do while true, criamos media, que é igual a soma / num, e 2 arrays, um chamado menor que 7 e outro maior que media, fazemos um for in de notas, checando se a nota é maior que a media, e se for, colocando no array, e se a nota é menor que 7, colocando no outro array. Ao fim do for in, criamos o array notas reverse, depois fazemos um for of de notas.reverse( ), para pegar as notas ao contrario, dentro do for, guardamos cada variável em um novo array, junto com um \n, para exibir ele de cima para baixo. No alert, surge um novo metodo, o .join(““), que exibe a lista, porém sem as virgulas entre seus elementos.



16- Utilize uma lista para resolver o problema a seguir. Uma empresa paga seus vendedores com base em comissões. O vendedor recebe $200 por semana mais 9 por cento de suas vendas brutas daquela semana. Por exemplo, um vendedor que teve vendas brutas de $3000 em uma semana recebe $200 mais 9 por cento de $3000, ou seja, um total de $470. Escreva um programa (usando um array de contadores) que determine quantos vendedores receberam salários nos seguintes intervalos de valores:

$200 - $299

$300 - $399

$400 - $499

$500 - $599

$600 - $699

$700 - $799

$800 - $899

$900 - $999

$1000 em diante

Desafio: Crie uma fórmula para chegar na posição da lista a partir do salário, sem fazer vários ifs aninhados.

[LINK](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/trunc) ou https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\_Objects/Math/trunc

Criamos um array e uma variável chamada salário. Faça um while true, pedindo a venda bruta, se a venda bruta foi igual a -1, termine o while com break, para o calculo do salario, passaremos que salario é igual a math.truc((200 + vendedor/100\*9)/100), o que significa que o valor da venda bruta dividido por 100, vezes 9, mais 200, divido por 100, que retorna um numero entre 1 a 10, math.trunc retorna apenas o inteiro daquele numero, ou seja. Se for 1.2, a resposta fica 1. Fazemos um if para checar se salario é maior que 10, se sim, salario = 10, após isso, adicionamos esse valor no array de salários. Ao final, faremos um alert checando com filter o numero de aparições de cada numero da lista até 10.



17- Em uma competição de salto em distância cada atleta tem direito a cinco saltos. O resultado do atleta será determinado pela média dos cinco valores restantes. Você deve fazer um programa que receba o nome e as cinco distâncias alcançadas pelo atleta em seus saltos e depois informe o nome, os saltos e a média dos saltos. O programa deve ser encerrado quando não for informado o nome do atleta. A saída do programa deve ser conforme o exemplo abaixo:

Atleta: Rodrigo Curvêllo

Primeiro Salto: 6.5 m

Segundo Salto: 6.1 m

Terceiro Salto: 6.2 m

Quarto Salto: 5.4 m

Quinto Salto: 5.3 m

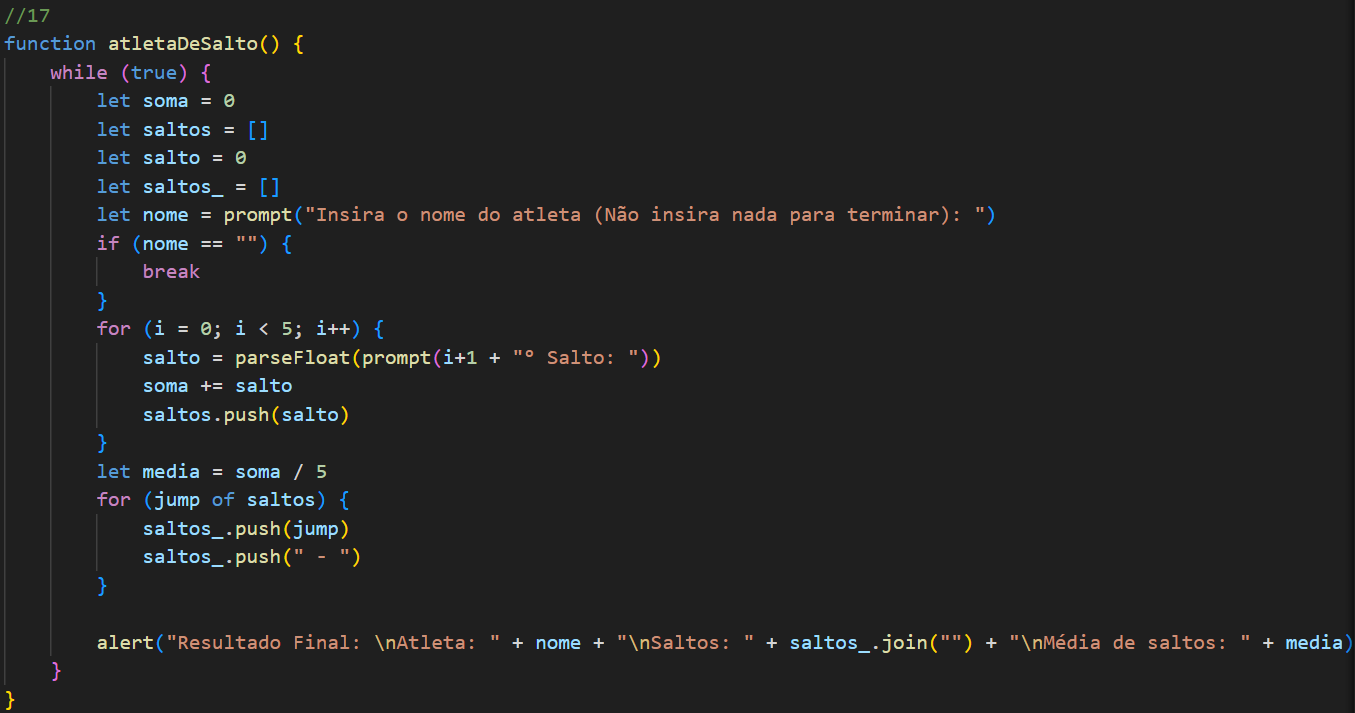
Resultado final:

Atleta: Rodrigo Curvêllo

Saltos: 6.5 - 6.1 - 6.2 - 5.4 - 5.3

Média dos saltos: 5.9 m

Essa atividade começa com a criação de um while true, dentro dele, declaramos as variáveis soma e salto como 0, saltos e saltos\_ como arrays vazios. Declaramos que nome é um prompt, para pedir ao cliente o nome. Fiz um if para se nome não foi preenchido, ele fecha o while, cria-se um for de 5 repetições pedindo o valor do salto com um prompt, fazemos a soma dentro da variável soma e guardamos salto no array de saltos. Após isso, criamos a variável media e fazemos a media dividindo por cinco a soma. Abrimos um for of dos saltos, para guarda-los em outra lista, a lista saltos\_, depois, colocamos tudo no alert.



18- Uma grande emissora de televisão quer fazer uma enquete entre os seus

telespectadores para saber qual o melhor jogador após cada jogo. Para isto, faz-se necessário o desenvolvimento de um programa, que será utilizado pelas telefonistas, para a computação dos votos.

Sua equipe foi contratada para desenvolver este programa, utilizando a linguagem de programação C++. Para computar cada voto, a

telefonista digitará um número, entre 1 e 23, correspondente ao número da camisa do jogador. Um número de jogador igual zero, indica que a votação foi encerrada. Se um número inválido for digitado, o programa deve ignorá-lo, mostrando uma breve mensagem de aviso, e voltando a pedir outro número. Após o final da votação, o programa deverá exibir:

O total de votos computados;

Os números e respectivos votos de todos os jogadores que receberam votos;

O percentual de votos de cada um destes jogadores;

O número do jogador escolhido como o melhor jogador da partida, juntamente com o número de votos e o percentual de votos dados a ele.

Observe que os votos inválidos e o zero final não devem ser computados como votos. O resultado aparece ordenado pelo número do jogador. O programa deve fazer uso de arrays. O programa deverá executar o cálculo do percentual de cada jogador através de uma função. Esta função receberá dois parâmetros: o número de votos de um jogador e o total de votos. A função calculará o percentual e retornará o valor calculado. Abaixo segue uma tela de exemplo. A disposição das informações deve ser a mais próxima possível ao exemplo. Os dados são fictícios e podem mudar a cada execução do programa. Ao final, o programa deve ainda gravar os dados referentes ao

resultado da votação em um arquivo texto no disco, obedecendo a mesma disposição apresentada na tela.

Enquete: Quem foi o melhor jogador?

Número do jogador (0=fim): 9

Número do jogador (0=fim): 10

Número do jogador (0=fim): 9

Número do jogador (0=fim): 10

Número do jogador (0=fim): 11

Número do jogador (0=fim): 10

Número do jogador (0=fim): 50

Informe um valor entre 1 e 23 ou 0 para sair!

Número do jogador (0=fim): 9

Número do jogador (0=fim): 9

Número do jogador (0=fim): 0

Resultado da votação:

Foram computados 8 votos.

Jogador Votos %

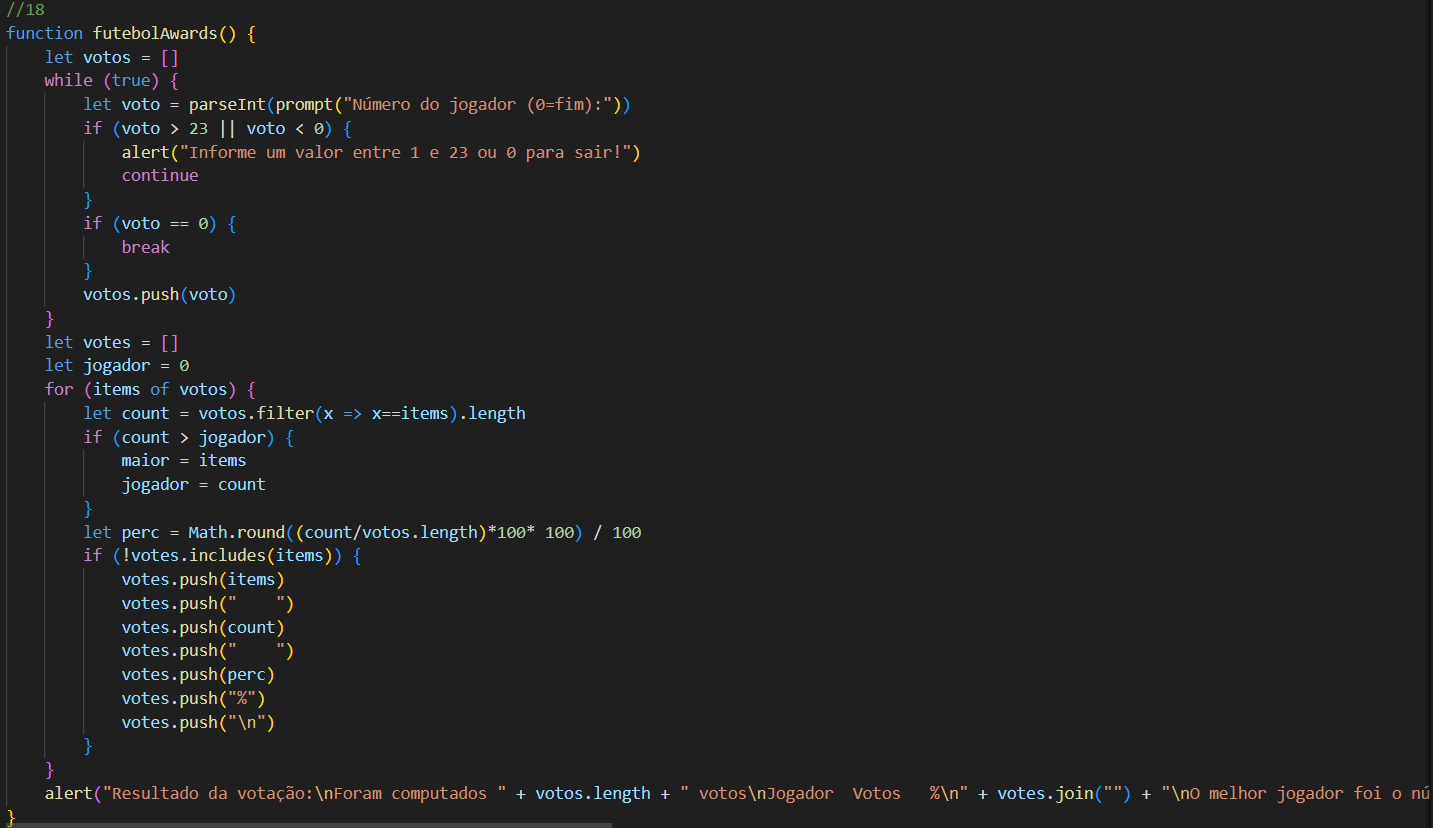
9 4 50,0%

10 3 37,5%

11 1 12,5%

O melhor jogador foi o número 9, com 4 votos, correspondendo a 50% do total de votos.

Comece criando o array votos, um while true, dentro dele, declaro a variável voto como um pedido, faço um check se o numero é maior que 23 ou menor que 0, se sim, ele continua para o próximo loop de repetição após o alerta de um erro, se o valor de voto foi igual a 0, ele termina o loop. Se não, ele coloca a variável voto dentro do array de votos. Ao fim do while, declaro um array chamado votes, uma variável chamada jogador igual a 0 e início um for of de votos, dentro dele, eu crio a variável count que guarda o numero de vezes que o item aparece na lista votos, depois, faço um if para ver se o numero de vezes é maior que jogador, se sim, a variável maior vai ser igual a items, e a variável jogador igual a count, isso será usado mais para frente para dizer qual o jogador que tem o maior numero de aparições, em seguida, fazemos o calculo da porcentagem de cada item. Por fim, fazemos uma checagem para se na lista votes, está incluso o item, se não, ele adiciona na lista o item, o numero de vezes que ele aparece e a porcentagem. Finalmente criamos um alert para todas essas informações da forma como foi pedido.



19- Uma empresa de pesquisas precisa tabular os resultados da seguinte enquete feita a uma grande quantidade de organizações:

"Qual o melhor Sistema Operacional para uso em servidores?"

As possíveis respostas são:

1- Windows Server

2- Unix

3- Linux

4- Netware

5- Mac OS

6- Outro

Você foi contratado para desenvolver um programa que leia o resultado da enquete e informe ao final o resultado da mesma. O programa deverá ler os valores até ser informado o valor 0, que encerra a entrada dos dados. Não deverão ser aceitos valores além dos válidos para o programa (0 a 6). Os valores referentes a cada uma das opções devem ser armazenados num vetor. Após os dados terem sido completamente informados, o programa deverá calcular a percentual de cada um dos concorrentes e informar o vencedor da enquete. O formato da saída foi dado pela empresa, e é o

seguinte:

Sistema Operacional Votos %

------------------- ----- ---

Windows Server 1500 17%

Unix 3500 40%

Linux 3000 34%

Netware 500 5%

Mac OS 150 2%

Outro 150 2%

------------------- -----

Total 8800

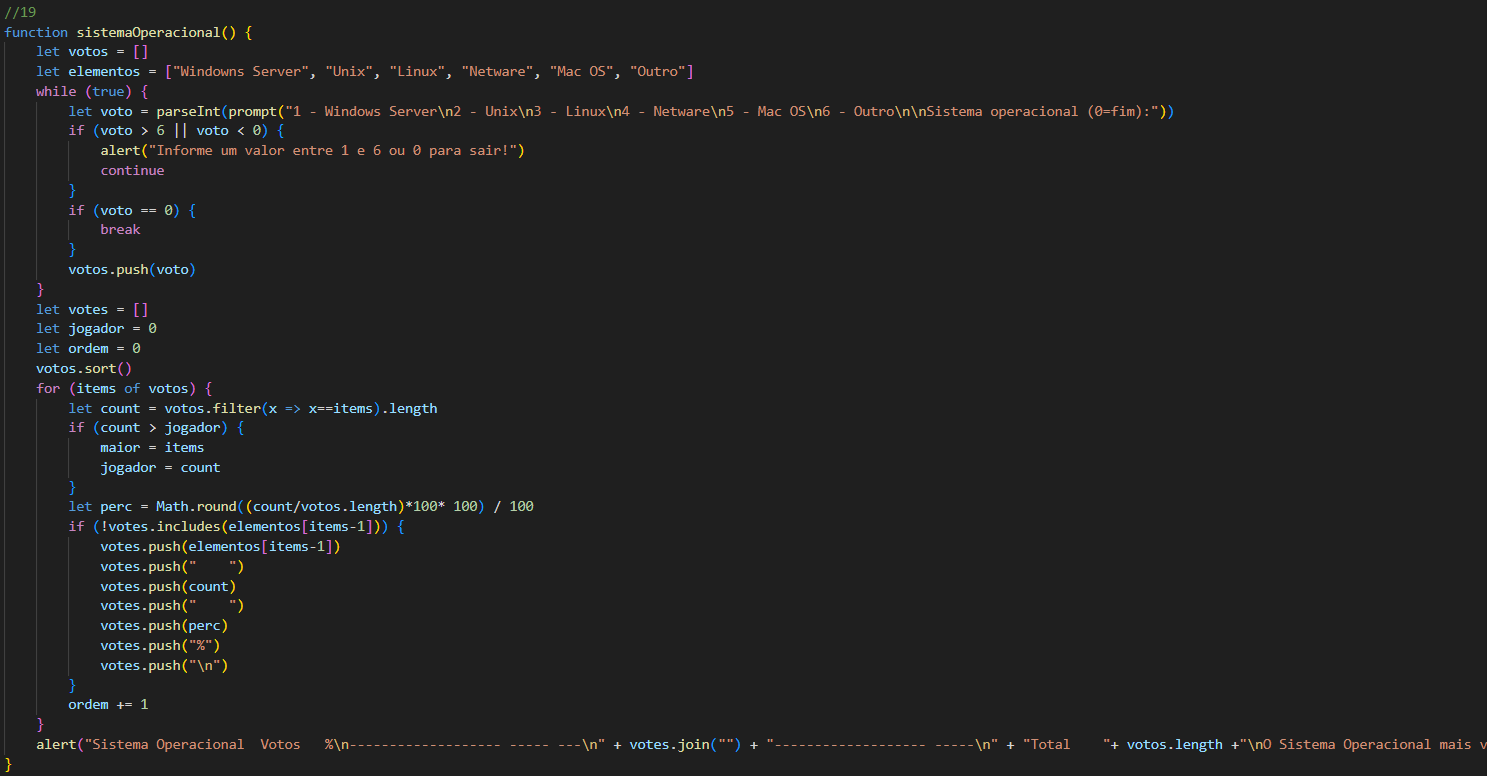
O Sistema Operacional mais votado foi o Unix, com 3500 votos, correspondendo a 40%

dos votos.

[LINK](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/round) ou <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/round>

Nessa atividade, MUITO SEMELHANTE A ANTERIOR, a única diferença da explicação anterior é a criação de uma lista com o nome dos elementos, o que provoca alguns problemas e soluções de erros da atividade anterior, por exemplo, na checagem se o item pertence a votes, se o count tiver o mesmo valor do adicionado, ele não exibe o valor, o que muda ao checar se o elemento na posição item + 1 está la. Como a atividade é idêntica, inclusive os nomes, por motivos óbvios, vou copiar a explicação e as coisas que mudei vou destacar com negrito. Outro comentário, o método round funciona da seguinte forma, ele arredonda para o numero inteiro mais próximo para baixo, nessa atividade, ele faz a conta da divisão, encontrando um decimal, multiplica por 100 para encontrar a porcentagem, vezes 100 de novo para puxar 2 numeros da virgula, em seguida arredondamos, tirando os outros números, em seguida, dividimos por 100, para encontrar novamente como porcentagem.

Comece criando o array votos **e um de elementos, contendo o nome de cada sistema operacional**, um while true, dentro dele, declaro a variável voto como um pedido, faço um check se o numero é maior que **6** ou menor que 0, se sim, ele continua para o próximo loop de repetição após o alerta de um erro, se o valor de voto foi igual a 0, ele termina o loop. Se não, ele coloca a variável voto dentro do array de votos. Ao fim do while, declaro um array chamado votes, uma variável chamada jogador igual a 0, **outra chamada ordem também igual a 0** e início um for of de votos, dentro dele, eu crio a variável count que guarda o numero de vezes que o item aparece na lista votos, depois, faço um if para ver se o numero de vezes é maior que jogador, se sim, a variável maior vai ser igual a items, e a variável jogador igual a count, isso será usado mais para frente para dizer qual o jogador que tem o maior numero de aparições, em seguida, fazemos o calculo da porcentagem de cada item. Por fim, fazemos uma checagem para se na lista votes, **está incluso o elemento na posição item, se não, ele adiciona na lista o elemento na posição item**, o numero de vezes que ele aparece e a porcentagem, **depois ordem += 1** . Finalmente criamos um alert para todas essas informações da forma como foi pedido.



20- As Organizações Tabajara resolveram dar um abono aos seus colaboradores em reconhecimento ao bom resultado alcançado durante o ano que passou. Para isto contratou você para desenvolver a aplicação que servirá como uma projeção de quanto será gasto com o pagamento deste abono. Após reuniões envolvendo a diretoria executiva, a diretoria financeira e os representantes do sindicato laboral, chegou-se a seguinte forma de cálculo:

a.Cada funcionário receberá o equivalente a 20% do seu salário bruto de dezembro;

a.O piso do abono será de 100 reais, isto é, aqueles funcionários cujo salário for muito baixo, recebem este valor mínimo; Neste momento, não se deve ter nenhuma preocupação com colaboradores com tempo menor de casa, descontos, impostos ou outras particularidades. Seu programa deverá permitir a digitação do salário de um número indefinido (desconhecido) de salários. Um valor de salário igual a 0 (zero) encerra a digitação. Após a entrada de todos os dados o programa deverá calcular o valor do abono concedido a cada colaborador, de acordo com a regra definida acima.

Ao final, o programa deverá apresentar:

O salário de cada funcionário, juntamente com o valor do abono;

O número total de funcionário processados;

O valor total a ser gasto com o pagamento do abono;

O número de funcionário que receberá o valor mínimo de 100 reais;

O maior valor pago como abono; A tela abaixo é um exemplo de execução do

programa, apenas para fins ilustrativos. Os valores podem mudar a cada execução do programa.

Projeção de Gastos com Abono

============================

Salário: 1000

Salário: 300

Salário: 500

Salário: 100

Salário: 4500

Salário: 0

Salário - Abono

R$ 1000.00 - R$ 200.00

R$ 300.00 - R$ 100.00

R$ 500.00 - R$ 100.00

R$ 100.00 - R$ 100.00

R$ 4500.00 - R$ 900.00

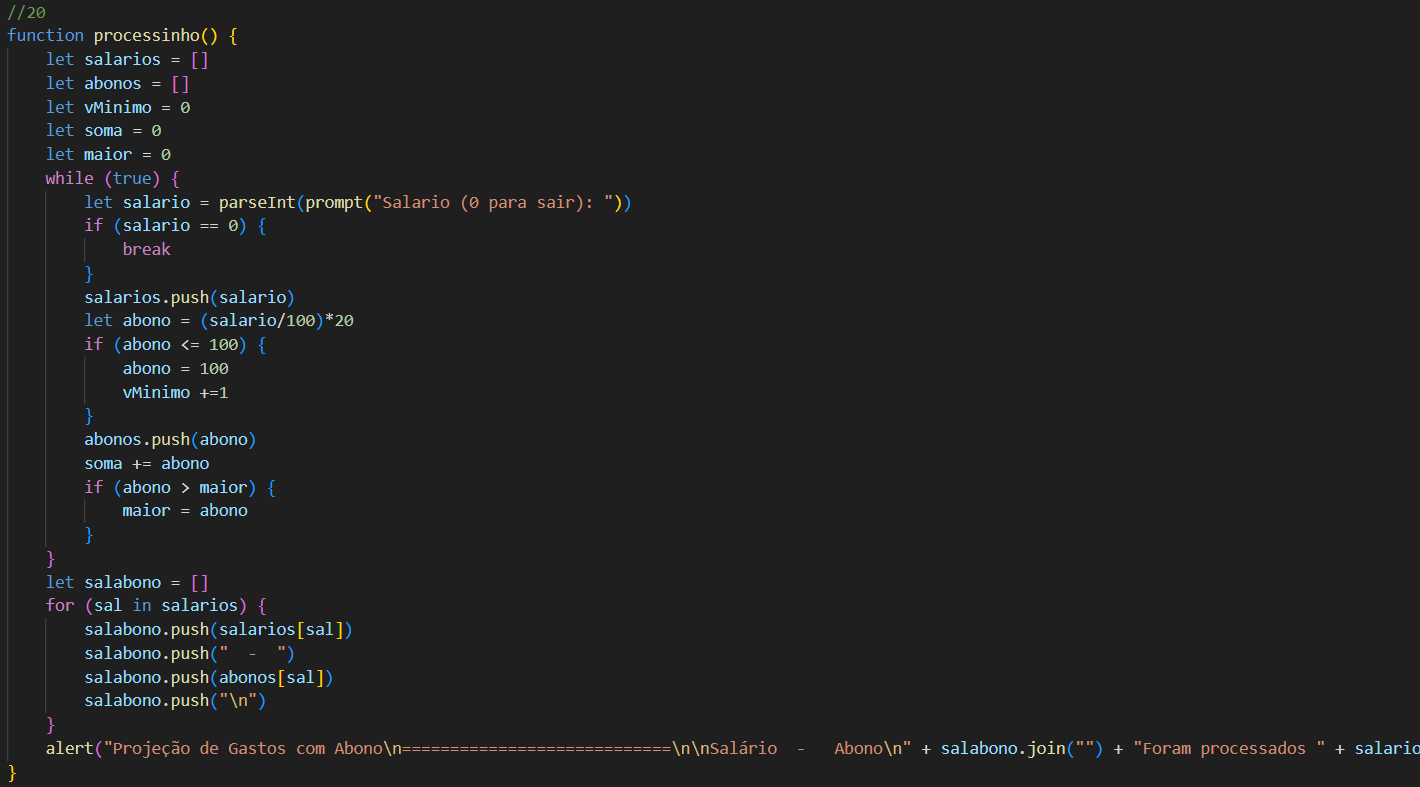
Foram processados 5 colaboradores

Total gasto com abonos: R$ 1400.00

Valor mínimo pago a 3 colaboradores

Maior valor de abono pago: R$ 900.00

Nessa atividade, começo criando 5 variaveis, um array de salários, um de abonos, um inteiro vMinimo igual a 0, soma também 0, e maior também. Um while true para realizar o pedido de salários, verificando também se salario for igual a 0, para brecar o while, fiz o cálculo do abono, após isso, um check se abono é menor ou igual a 100, se sim, salvamos abono como 100 e adicionamos mais um em vMinimo, para contar futuramente a quantidade de abonos mínimos. Em seguida salvamos o abono no array de abonos, colocamos abono na soma, checamos o maior abono. No fim do while, crio o array salabono, para incluir, semelhante as de cima, em um alert só todas as informações.



21- Faça um programa que carregue uma lista com os modelos de cinco carros (exemplo

de modelos: FUSCA, GOL, VECTRA etc). Carregue uma outra lista com o consumo desses carros, isto é, quantos quilômetros cada um desses carros faz com um litro de combustível. Calcule e mostre:

O modelo do carro mais econômico;

Quantos litros de combustível cada um dos carros cadastrados consome para

percorrer uma distância de 1000 quilômetros e quanto isto custará, considerando um que a gasolina custe R$ 2,25 o litro. Abaixo segue uma tela de exemplo. A disposição

das informações deve ser a mais próxima possível ao exemplo. Os dados são fictícios e podem mudar a cada execução do programa.

Comparativo de Consumo de Combustível

Veículo 1

Nome: fusca

Km por litro: 7

Veículo 2

Nome: gol

Km por litro: 10

Veículo 3

Nome: uno

Km por litro: 12.5

Veículo 4

Nome: Vectra

Km por litro: 9

Veículo 5

Nome: Peugeout

Km por litro: 14.5

Relatório Final

1 - fusca - 7.0 - 142.9 litros - R$ 321.43

2 - gol - 10.0 - 100.0 litros - R$ 225.00

3 - uno - 12.5 - 80.0 litros - R$ 180.00

4 - vectra - 9.0 - 111.1 litros - R$ 250.00

5 - peugeout - 14.5 - 69.0 litros - R$ 155.17

O menor consumo é do peugeout.

Crie diversas variáveis, um array de carros, um de consumos por litro, uma variável de carro econômico sem nenhuma informação, assim como de carro e consumo por litro, um array de litros por mil km, um array de preço, e uma variável chamada menor com um numero bem alto. Criei um for de 5 numeros para pedir o nome do carro, depois seu km por litro, coloquei-os em suas devidas listas. Em seguida, um for in da lista de consumos por litro, usando isso para checar qual o menor consumo por litro e guarda-lo junto com seu respectivo carro. Coloco no array de consumo por mil km o valor do consumo a cada mil km, depois no array de preço, seu preço respectivo em 1000 km. Em seguida, para organizar o alert, como nas outras, coloco todas as informações em uma lista e dou um alert com lista.join(“”).



22- Sua organização acaba de contratar um estagiário para trabalhar no Suporte de Informática, com a intenção de fazer um levantamento nas sucatas encontradas nesta área. A primeira tarefa dele é testar todos os cerca de 200 mouses que se encontram lá, testando e anotando o estado de cada um deles, para verificar o que se pode aproveitar deles.

Foi requisitado que você desenvolva um programa para registrar este levantamento. O

programa deverá receber um número indeterminado de entradas, cada uma

contendo: um número de identificação do mouse o tipo de defeito:

necessita da esfera;

necessita de limpeza; a.necessita troca do cabo ou conector; a.quebrado ou inutilizado

Uma identificação igual a zero encerra o programa. Ao final o programa deverá emitir

o seguinte relatório:

Quantidade de mouses: 100

Situação Quantidade Percentual

1- necessita da esfera 40 40%

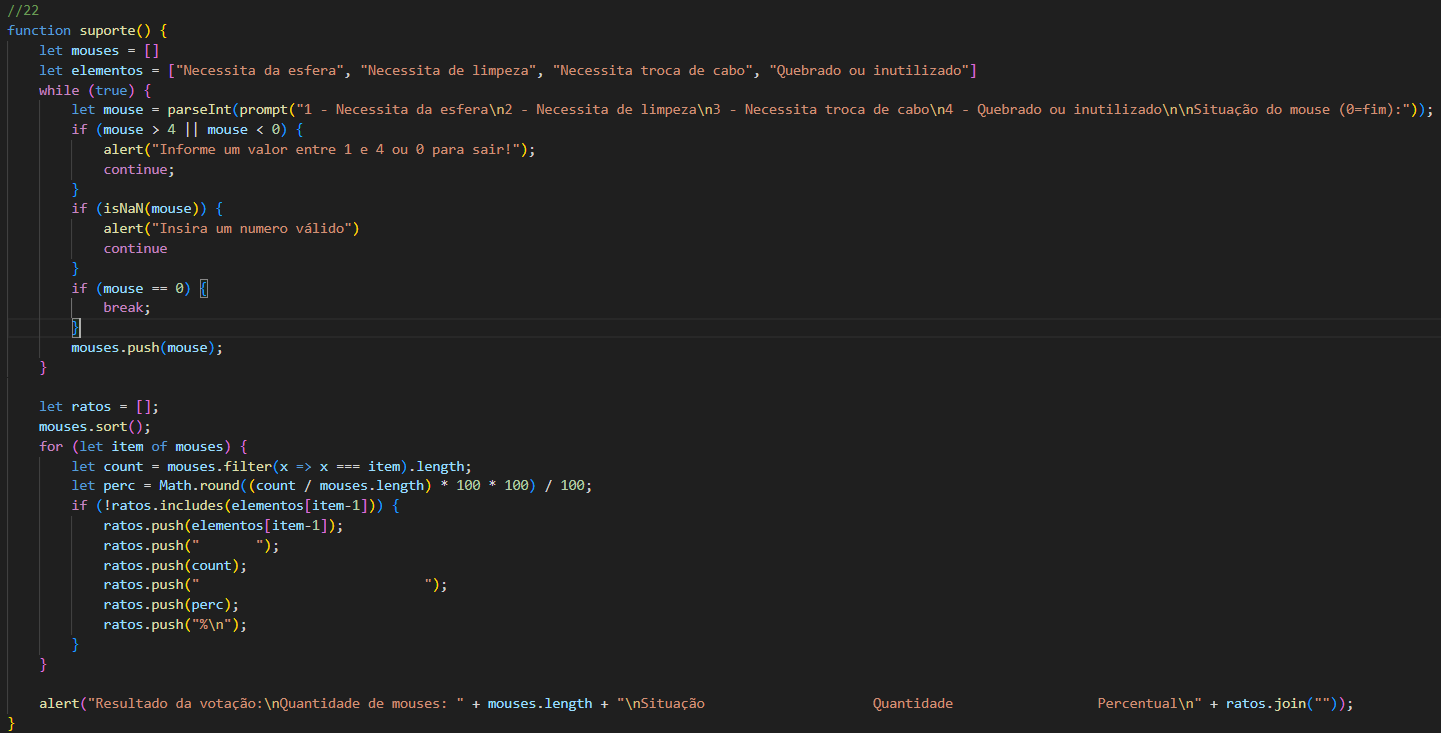
2- necessita de limpeza 30 30%

3- necessita troca do cabo ou conector 15 15%

4- quebrado ou inutilizado 15 15%

[LINK](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/isNaN) ou https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/isNaN

Começo com a criação de um array de mouses e um de elementos, este contendo o nome de todos os problemas possíveis do mouse. Depois um while true, pedindo o mouse, dando erro se ele for maior que 4 e menor que 0, outro se não for informado nada e para terminar o while se for 0, se nenhum if acontecer, ele apenas adiciona o mouse no array de mouses. Em seguida criei outro array, como anteriormente, para formatar o alert, esse chamado ratos( tradução de mouses ), em seguida, mouses . sort( ), que serve para organizar ele em ordem, para a exibição no alert. Faço um for of de mouses, com count para o numero de elementos, semelhante as anteriores, um de porcentagem, um if de não inclusos na lista, como anteriormente, e guardo tudo na lista, em seguida o alert com a lista e as outras informações.



23- A ACME Inc., uma empresa de 500 funcionários, está tendo problemas de espaço em disco no seu servidor de arquivos. Para tentar resolver este problema, o Administrador de Rede precisa saber qual o espaço ocupado pelos usuários, e identificar os usuários com maior espaço ocupado. Através de um programa, baixado da Internet, ele conseguiu gerar o seguinte arquivo, chamado "usuarios.txt":

alexandre 456123789

anderson 1245698456

antonio 123456456

carlos 91257581

cesar 987458

rosemary 789456125

Neste arquivo, o nome do usuário possui 15 caracteres. A partir deste arquivo, você deve criar um programa que gere um relatório, chamado "relatório.txt", no seguinte formato:

ACME Inc. Uso do espaço em disco pelos usuários

------------------------------------------------------------------------

Nr. Usuário Espaço utilizado % do uso

1 alexandre 434,99 MB 16,85%

2 anderson 1187,99 MB 46,02%

3 antonio 117,73 MB 4,56%

4 carlos 87,03 MB 3,37%

5 cesar 0,94 MB 0,04%

6 rosemary 752,88 MB 29,16%

Espaço total ocupado: 2581,57 MB

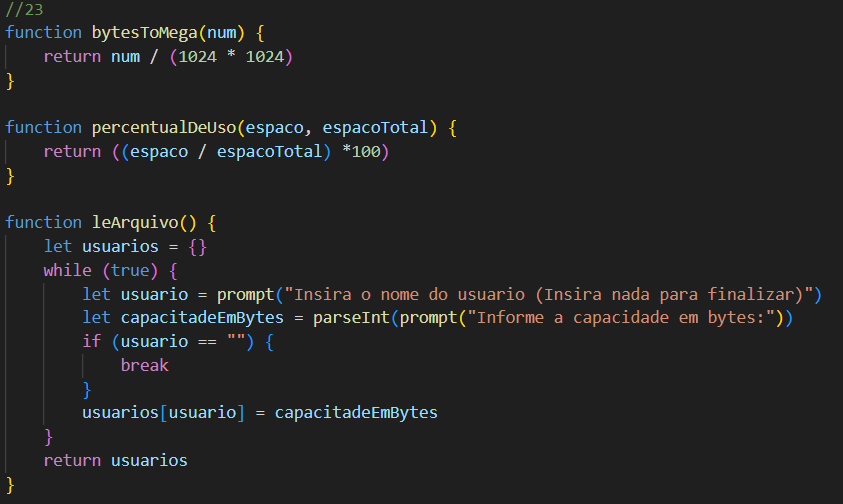
Espaço médio ocupado: 430,26 MB

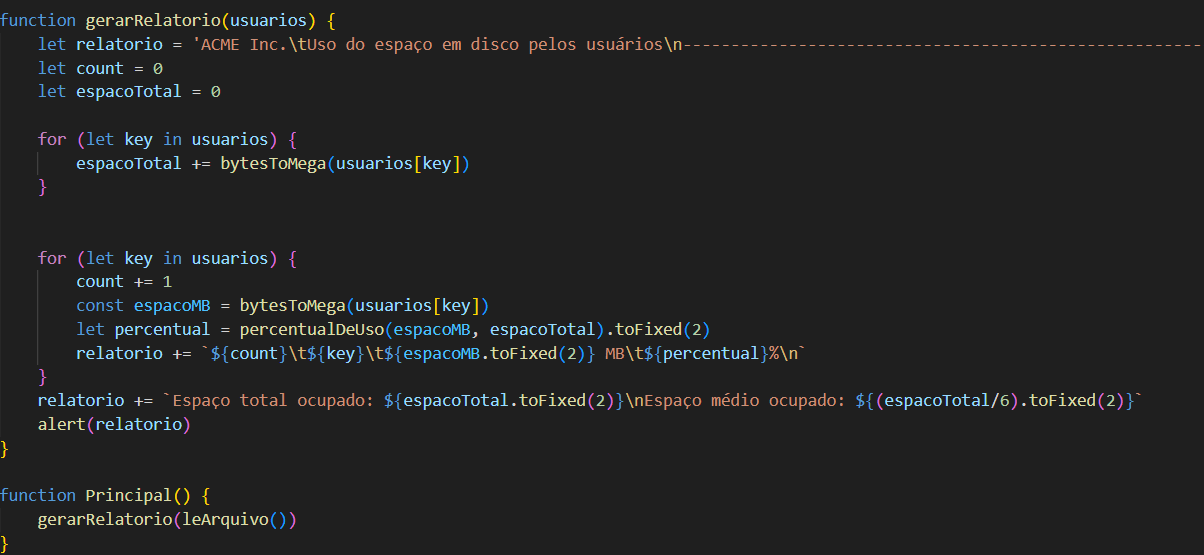
O arquivo de entrada deve ser lido uma única vez, e os dados armazenados em memória, caso sejam necessários, de forma a agilizar a execução do programa. A conversão do espaço ocupado em disco, de bytes para megabytes deverá ser feita através de uma função separada, que será chamada pelo programa principal. O cálculo do percentual de uso também deverá ser feito através de uma função, que será chamada pelo programa principal.

[LINK](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/return) ou https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/return

Começamos com separando as funções, a primeira a ser feita é a de bytes para megabytes, que simplesmente retorna o valor dado na função dividido por 1024 vezes 1024, em seguida outra função, de percentual de uso, pedindo 2 argumentos, retornando a divisão dos dois argumentos vezes 100. Em seguida, uma função para ler arquivos, que inicia declarando um map de usuários, um while true para pedir o nome do usuário, outro da capacidade em bytes, se o usuário não for preenchido, ele acaba o while, e por fim, completando usuários na key usuários, recebendo capacidade em bytes, no fim, retornando usuários. Outra função, para gerar relatórios, pedindo usuários como argumento ,que inicia declarando a variável relatório como um texto, a variável count como 0 e espaco total como 0, um for in de usuários somando para encontrar o espaço total passando a função bytes para megabytes no valor da key de usuários, em seguida outro for in de usuários, uma variável para contar quantos usuários tem, uma para a quantidade de megabytes e outro para o percentual de uso, em seguida, guardamos essas informações no relatório, concatenando isso na string principal do relatório.

Por fim, alertamos isso. Para fazer o código funcionar, criei uma função principal, a qual vai ser executada e executa a função gerar relatório passando a função le arquivos.





24- Faça um programa que simule um lançamento de dados. Lance o dado 100 vezes e armazene os resultados em um vetor. Depois, mostre quantas vezes cada valor foi conseguido. Dica: use um vetor de contadores (1-6) e uma função para gerar números aleatórios, simulando os lançamentos dos dados.

[LINK](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random) ou <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random>

Essa atividade é bem básica, a única coisa que precisamos é explicar como funciona o math.random() e o math.floor(), começando pelo math.random(), ele faz o seguinte, ele retorna um numero entre 1 e 0, passando para o math.floor(), ele faz um ajuste decimal, ou seja, ele retorna o valor inteiro de um decimal. Portanto essa atividade faz o seguinte, criamos uma função, que pede 2 argumentos, o máximo e o mínimo, com isso, fazemos um math.random dentro de um math.floor para encontrar um valor entre 0 e 1, e pegar o valor inteiro disso, no entanto, antes do math.floor ele multiplica o mínimo menos o máximo + 1, depois soma o mínimo, ou seja, ele encontra um valor entre 0 e 1, como 0.567, vezes 6 – 1 + 1, que é 6, ou seja 0.567 \* 6, mais 1, que da 4.402, dai math.floor, retornando 4, que foi o numero aleatório. Em seguida, criamos a função principal, que declara dados como array, depois um for de 100 repetições, declarando dado como a função de aleatórios com o intervalo de 1 e 6, em seguida guardamos isso em dados. No alert, apenas utilizamos filter para encontrar o numero de vezes que cada numero aparece e exibi-lo.



Versões alternativas – 18 e 23.

18- Uma grande emissora de televisão quer fazer uma enquete entre os seus

telespectadores para saber qual o melhor jogador após cada jogo. Para isto, faz-se necessário o desenvolvimento de um programa, que será utilizado pelas telefonistas, para a computação dos votos.

Sua equipe foi contratada para desenvolver este programa, utilizando a linguagem de programação C++. Para computar cada voto, a

telefonista digitará um número, entre 1 e 23, correspondente ao número da camisa do jogador. Um número de jogador igual zero, indica que a votação foi encerrada. Se um número inválido for digitado, o programa deve ignorá-lo, mostrando uma breve mensagem de aviso, e voltando a pedir outro número. Após o final da votação, o programa deverá exibir:

O total de votos computados;

Os números e respectivos votos de todos os jogadores que receberam votos;

O percentual de votos de cada um destes jogadores;

O número do jogador escolhido como o melhor jogador da partida, juntamente com o número de votos e o percentual de votos dados a ele.

Observe que os votos inválidos e o zero final não devem ser computados como votos. O resultado aparece ordenado pelo número do jogador. O programa deve fazer uso de arrays. O programa deverá executar o cálculo do percentual de cada jogador através de uma função. Esta função receberá dois parâmetros: o número de votos de um jogador e o total de votos. A função calculará o percentual e retornará o valor calculado. Abaixo segue uma tela de exemplo. A disposição das informações deve ser a mais próxima possível ao exemplo. Os dados são fictícios e podem mudar a cada execução do programa. Ao final, o programa deve ainda gravar os dados referentes ao

resultado da votação em um arquivo texto no disco, obedecendo a mesma disposição apresentada na tela.

Enquete: Quem foi o melhor jogador?

Número do jogador (0=fim): 9

Número do jogador (0=fim): 10

Número do jogador (0=fim): 9

Número do jogador (0=fim): 10

Número do jogador (0=fim): 11

Número do jogador (0=fim): 10

Número do jogador (0=fim): 50

Informe um valor entre 1 e 23 ou 0 para sair!

Número do jogador (0=fim): 9

Número do jogador (0=fim): 9

Número do jogador (0=fim): 0

Resultado da votação:

Foram computados 8 votos.

Jogador Votos %

9 4 50,0%

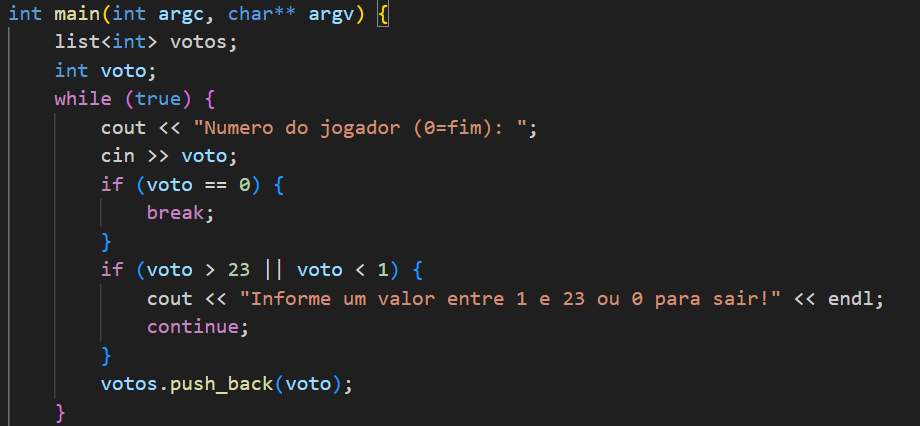
10 3 37,5%

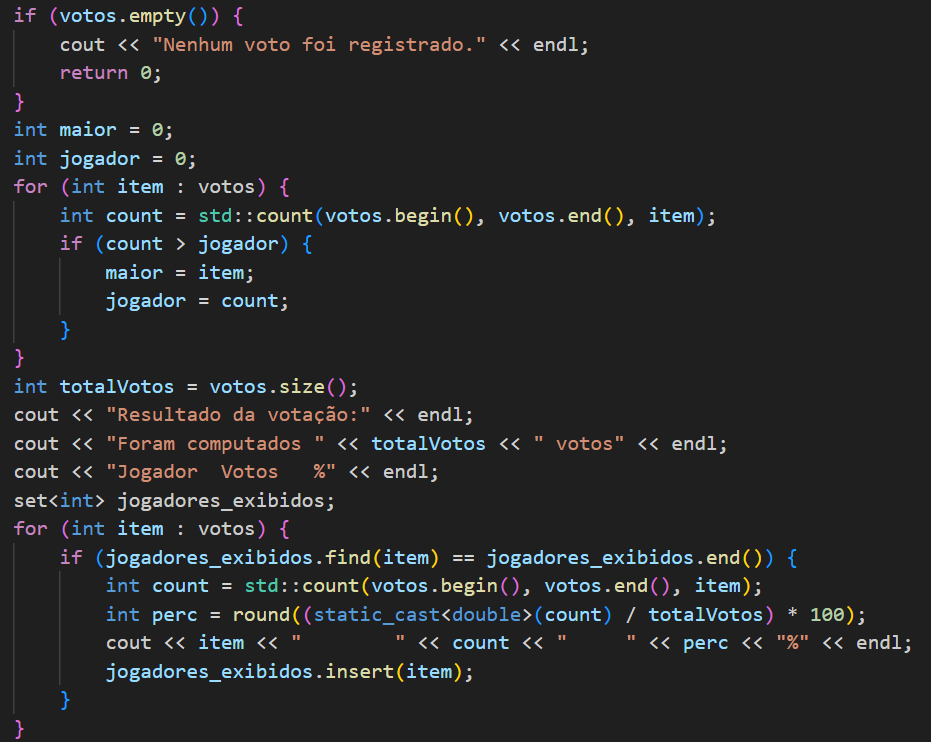
11 1 12,5%

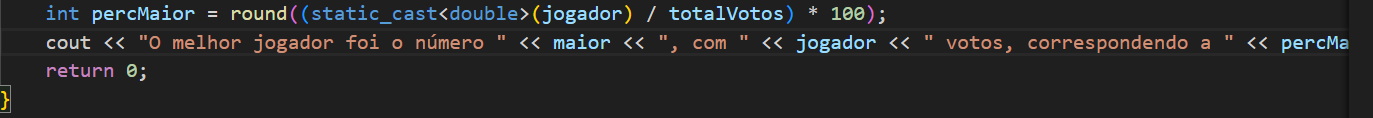
O melhor jogador foi o número 9, com 4 votos, correspondendo a 50% do total de votos.

Essa atividade alternativa foi feita em c++, conforme pedia a atividade.

Começamos passando a função principal, chamada main, declaramos uma lista de inteiros chamada votos, um int chamado voto, começamos o while true, passamos o pedido de numero do jogador e guardamos com cin >> na variável voto, fazemos um if para verificar se voto é igual a 0 e se sim, fechamos o while, depois outro if checando se o numero excede os valores, se sim, continue para continuar o while depois da mensagem de erro dada pelo cout, se não houver bloqueio nos ifs, o valor de voto vai para a lista de votos pelo push\_back, em seguida outro if checando se a lista votos está vazia, se sim, parando a função, se não, declaramos um inteiros chamado maior e outro chamado jogador, ambos valendo 0, fazemos um for of de votos e armazenamos a quantidade de vezes que esse valor aparece na lista com std:count(votos.begin(), votos.end(), item), que checa se do começo ao fim da lista, quandas vezes o item apareceu, depois disso um if para verificar o maior, se a quantidade for maior que jogador, ele declara maior como item e jogador como count. Depois um inteiro de totalVotos recebendo o tamanho da lista, depois um alerta para mostrar os resultados, um set<int> chamado jogadores exibidos, após isso um for para adicionar na lista nova chamada jogadores exibidos, dentro do for, uma checagem, se dentro da lista tiver o valor, ele não é adiciona, se não tiver, ele é adicionado, dentro do if, fazemos o numero de vezes que ele aparece, a porcentagem e juntamos isso em item e colocamos na lista. Após tudo isso, temos um calculo da maior porcentagem, que é feito da seguinte forma, trocamos o valor de lista de ints para doubles, e dividimos pelo total de votos e multiplicamos por cem e arredondamos com round para encontrar a porcentagem. Por fim, um cout para exibir todas as informações e retornando 0 pela necessidade disso em c++.







23- A ACME Inc., uma empresa de 500 funcionários, está tendo problemas de espaço em disco no seu servidor de arquivos. Para tentar resolver este problema, o Administrador de Rede precisa saber qual o espaço ocupado pelos usuários, e identificar os usuários com maior espaço ocupado. Através de um programa, baixado da Internet, ele conseguiu gerar o seguinte arquivo, chamado "usuarios.txt":

alexandre 456123789

anderson 1245698456

antonio 123456456

carlos 91257581

cesar 987458

rosemary 789456125

Neste arquivo, o nome do usuário possui 15 caracteres. A partir deste arquivo, você deve criar um programa que gere um relatório, chamado "relatório.txt", no seguinte formato:

ACME Inc. Uso do espaço em disco pelos usuários

------------------------------------------------------------------------

Nr. Usuário Espaço utilizado % do uso

1 alexandre 434,99 MB 16,85%

2 anderson 1187,99 MB 46,02%

3 antonio 117,73 MB 4,56%

4 carlos 87,03 MB 3,37%

5 cesar 0,94 MB 0,04%

6 rosemary 752,88 MB 29,16%

Espaço total ocupado: 2581,57 MB

Espaço médio ocupado: 430,26 MB

O arquivo de entrada deve ser lido uma única vez, e os dados armazenados em memória, caso sejam necessários, de forma a agilizar a execução do programa. A conversão do espaço ocupado em disco, de bytes para megabytes deverá ser feita através de uma função separada, que será chamada pelo programa principal. O cálculo do percentual de uso também deverá ser feito através de uma função, que será chamada pelo programa principal.

Para começar, essa atividade pede para que leiamos um txt, nesse sentido, uma obrigatoriedade de ler em web é a de usar em javascript as funções com async e await, que faz com que a resposta gere uma promisse, que quando houver necessidade de usar aquela resposta ele utilize essa promessa de uma resposta, então, quando essa resposta for lida, ele utiliza ela no restante do código. Para isso, as funções chamam async e use await no recebimento da variável quando for utilizar algo desse padrão de comportamento.

Portanto começaremos a explicação principal. Primeiro foi feita a função de ler o arquivo, que declara response como um fetch do arquivo(argumento da função), fetch serve para ler um arquivo txt, conforme pede a atividade, esse arquivo foi escrito em formato JSON, por mais que seja um txt, pela facilidade, em seguida, transformamos esse fetch do arquivo resposta em um json e retornamos ele. Em seguida outra função igual a da atividade normal, de bytes para mega, e também a função de percentual, totalmente igual. Entretanto, na função de gerar relatório, há algo diferente, ela é um async function, por conta de receber usuarioJSON. Nela fazemos o mesmo da atividade anterior, tirando que usaremos o document,getelementbyid para colocar o resultado de relatório em um id no html, assim sem precisar de um alert. Por ultimo, colocamos tudo na função principal e rodamos essa função.

