Explicação de resoluções de exercícios de Javascript

Othavio Henrique Melancieri Amaral

1 — Simplesmente decladei uma variável vetor e fiz ela aparecer como saída usnado o comando “alert()”, a soma dela com uma *string* foi para concatenar, já que “alert()” aceita apenas um argumento, isso tudo foi aprendido por meio de aulas do curso Técnico em Desenvolvimento de Sistemas. Explicação sobre o que é um *array* e vetores:

- https://dicasdeprogramacao.com.br/o-que-sao-vetores-e-matrizes-arrays/

2 — Número reais eu aprendi nas aulas de matemática do Ensino Fundamental II, para inverter a lista eu usei “reverse()”, encontrei a função por meio das sugestões do “VS Code”, eu usei o atalho “ctrl + espaço” e apareceu esta função, então testei ela e vi que ela fazia realmente o que eu esperava.

3 — O comando “prompt()” foi aprendido em aulas do curso Técnico em Desenvolvimento de Sistemas por meio de explicações do professor, média aritmética eu aprendi em aulas de matemática da escola SESI Samir Nakad. Eu somei o valor calculado a uma *string* para funcionar no momento de saída, já que o “alert()” aparentemente aceita apenas um argumento.

4 — Sites usados para descobrir como contar consoantes:

- https://www.geeksforgeeks.org/find-number-of-consonants-in-a-string-using-javascript/
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/match

Eu percebi que o método “match()” vê o que tem nele como argumento da função e em uma *string* que está sendo verificada, o que têm nos dois é então retornado em uma lista, é possível saber a quantidade de dígitos que combinaram extraindo o comprimento dela por meio de “.length”.

5 — O conhecimento lógico foi pensado facilmente, já que em aulas anteriores do curso Técnico de Desenvolvimento de Sistemas, um exercício similar já foi feito. Aqui foram usados apenas funções padrões e métodos já mostrados antes em aula.

6 — Para este exercício eu usei minha própria lógica, me baseando no que li no livro “Algoritmos Lógica para Desenvolvimento de Programação de Computadores” por José Augusto N. G. Manzano e Jayr Figueiredo de Oliveira, no capítulo 8: “Estruturas de Dados Homogêneas de Duas Dimensões”, já que eu necessitei de uma matriz bidimensional. Usei a variável “contador” somente para iterar mais facilmente no laço “for” aninhado já que eu ainda não entendi como funciona o laço no Javascript em sua versão clássica.

7 — Embora eu acho que devam existir funções predefinidas do Javascript para isso como a “sum()” em *Python*, eu optei por fazer manualmente mesmo, iterando através de um loop “for” para calcular, precisei das variáveis “mult” para guardar o resultado da multiplicação e “soma” para o da soma.

8 — Eu tentei seguir a mesma lógica usada no exercício 6, porém desta vez para adicionar valores ao invés de verificar eles, só que isso deu errado, então usei dois laços “for” sem estarem aninhados, a variável “contador” serve somente para mostrar uma mensagem mais descritiva ao usuário, para ele saber o valor de qual pessoa ele está a inserir. Meu algoritmo inteiro se divide em duas partes, um primeiro laço que recebe os valores altura e idade respectivamente e os insere em listas menores dentro da lista principal que contém todas as pessoas, o segundo laço é somente para percorrer tudo depois e mostrar a saída desses dados em ordem invertida. Este exercício em particular levou mais tempo para eu desenvolver a lógica do que outros. Site que eu visitei enquanto desenvolvia a lógica: (eu tentei usar um laço “for” clássico também, porém não consegui o encaixar na lógica).

- https://www.w3schools.com/js/js_loop_for.asp

9 — Para este exercício eu tive que aprender um pouco sobre o escopo das variáveis criadas pelas palavras-chave “var” e “let”, juntando meu conhecimento que eu já tinha adquirido nas aulas e na pesquisa anterior do exercício 8, naquele site eu consegui entender claramente o uso do laço “for” clássico no Javascript. Outro site consultado: (além do W3schools que teve significativa contribuição).

- https://www.alura.com.br/artigos/entenda-diferenca-entre-var-let-e-const-no-javascript?utm_term=&utm_campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conte%C3%BAdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=164068847699&hsa_ad=703934793962&hsa_src=g&hsa_tgt=dsa-2273097816642&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=Cj0KCQjw0Oq2BhCCARIsAA5hubWjgR0ev0k1nv3VBX1B_33q7TALasQhjEWvD3Tvk8UfaUE6Ld7sEB4aAslPEALw_wcB

10 — Neste exercício foi de novo só utilizar o que aprendi pesquisando com exercícios de antes, mais especificamente sore o escopo de variáveis geradas com “let” e “var” e como usar o laço “for” clássico (fontes de pesquisa já listadas antes). Aqui meu algoritmo se divide em declarações iniciais e três laços principais, primeiro declarei os três vetores vazios que seriam enchidos com valores inseridos pelo usuário mais tarde; no primeiro laço, são coletadas entradas do usuário e inseridas no vetor 1; no segundo laço a mesma coisa, mas com o vetor 2; já no terceiro é tudo intercalado dentro do

terceiro vetor, e, finalmente então, ocorre a saída de dados por meio da função “alert()” com o terceiro vetor.

11 — Foi só modificar o exercício 10, adicionei mais um vetor e um laço para ele, agora o programa declara 4 vetores vazios iniciais ao invés de 3, o vetor 3 agora segue a mesma lógica do 1 e do 2, enquanto isso o novo vetor 4 é o que recebe todos os valores intercalados com o último laço, para adicionar um terceiro vetor a ser intercalado neste laço eu só tive que adicionar uma linha, isso significa que agora a cada iteração do laço final são inseridos 3 valores dentro do vetor intercalado ao invés de 2.

12 — Para fazer este exercício eu me baseei na lógica do exercício 6 (considere a fonte de pesquisa usada na atividade 12 a mesma da 6), usei uma matriz que contém os alunos, cada aluno contém dois valores que são respectivamente idade e altura, comecei declarando a matriz já com os valores simulados dentro dela, logo depois declarei três variáveis para cálculo e contabilização (“maisDe13MaisBaixoQueMedia”, “soma” e “media”), depois disso o algoritmo executa o primeiro laço, que serve somente para acessar o valor de altura de cada aluno e somá-lo, e então é calculado a média de altura; depois começa o segundo laço, que contabiliza o total de alunos que têm idade maior que 13 anos e altura menor que a média, a saída só vem no final com a função “alert()” igual a todos os exercícios.

13 — Nada que eu precise explicar neste exercício que não seja a lógica, já que a maneira de se lidar com listas e condicionais foi aprendida parte em aula e outra parte enquanto pesquisava para outros exercícios. Primeiramente eu criei uma variável para coletar o valor de temperatura média de cada mês, depois dessas variáveis veio outras: duas listas, uma *string* vazia e dois valores numéricos iguais a 0 para cálculo (“soma” e “media”), então eu coloquei todos os valores de temperatura média dos meses na lista “ano”, depois eu fiz um laço “for” do estilo “posição em lista” para calcular a soma por meio do acesso ao valor de cada mês, calculei a média somente após o laço para economizar processamento, depois de tudo isso vem o segundo laço em que o algoritmo é dividido, para cada mês ele verifica se seu valor é maior que a média, se for, ele então verifica o índice da lista para mais tarde mandar uma *string* específica para a lista secundária que vai ser então usada na saída no fim do algoritmo.

14 — Neste exercício eu comecei declarando duas listas, uma com as classificações e outra com as perguntas, declarei uma outra variável para contabilizar (por isso o valor dela começa como 0), depois fiz um comando de saída só para o usuário saber o que ele está prestes a responder, e assim começam as perguntas, para cada “s” que o usuário responde é contabilizado uma resposta como “sim”, depois é feita uma outra seção de verificações, em que é verificada a quantidade de questões respondidas com “s”, isso para decidir a classificação da pessoa interrogada no final. Todos as perguntas e classificações estão em listas e são usadas em comandos “prompt()” e “alert()”. Site usado para descobrir o método “toLowerCase()”:

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/toLowerCase

15 — Comecei declarando variáveis vazias para usar depois, cinco para contabilizar e uma para guardar coisas (uma lista), depois vêm a parte principal do programa, o laço "while", que vi dentro deste site por meio de pesquisa: https://www.w3schools.com/js/js_loop_while.asp , o comando "break" eu aprendi porque um professor do curso Técnico em Desenvolvimento de Sistemas me falou sobre e eu já sabia que o *Python* e o *Luau* o tinham também, as notas sobem para a lista se não forem "-1", depois tudo que precisa ser contabilizado é feito por seus correspondentes laços "for" e variáveis, a saída é feita de maneira ordenada com múltiplas funções "alert()".

16 — Eu demorei um pouco para desvendar como deveria ser a lógica deste exercício, nele não havia especificações de como deveriam entrar os valores, então resolvi fazer um laço de entrada que se encerra com "-1" como resposta do usuário, mas antes eu declarei duas listas: uma para conter a quantidade de pessoas em cada intervalo, e a outra para os valores recebidos que vai aumentando conforme as entradas do usuário vão chegando através do laço. Depois do laço de entrada há um laço "for" que checa todos os valores recebidos do usuário e aumenta os valores padrões dentro da lista que conta quantas pessoas há em cada intervalo conforme vai verificando e decidindo. No final há a saída, que usa os próprios valores dentro da lista contadora.

17 — Aqui usei um laço "for" clássico e conhecimento sobre escopo, antes aprendidos respectivamente em:

- https://www.w3schools.com/js/js_loop_for.asp
- https://www.alura.com.br/artigos/entenda-diferenca-entre-var-let-e-const-no-javascript?utm_term=&utm_campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conte%C3%BAdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=164068847699&hsa_ad=703934793962&hsa_src=g&hsa_tgt=dsa-2273097816642&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=Cj0KCQjw0Oq2BhCCARIsAA5hubWjgR0ev0k1nv3VBX1B_33q7TALasQhjEWvD3Tvk8UfaUE6Ld7sEB4aAslPEALw_wcB

mas primeiro, meu código começa pedindo o nome do jogador, se não houver resposta, nada acontece; mas se houver, o código cria a lista que vai guardar os valores em metros dos saltos do jogador e inicia o primeiro laço, em que é recebido e adicionado à lista os valores de cada salto. Depois é inicializada uma variável para cálculo chamada "soma", que eu criei usando "let" para economizar memória já que ela é somente para processamento e não necessita de um escopo global (e acabei fazendo o mesmo com

quase todas as variáveis, já que o código se baseia dentro de um bloco condicional), a variável “soma” é usada em um laço “for” para calcular a média, que então é usada na saída junto com os resultados em uma função “alert()”.