Atividades JavaScript

Raul Elias Barbosa Rodrigues

Curso Técnico de desenvolvimento de sistemas

06/ setembro/ 2024

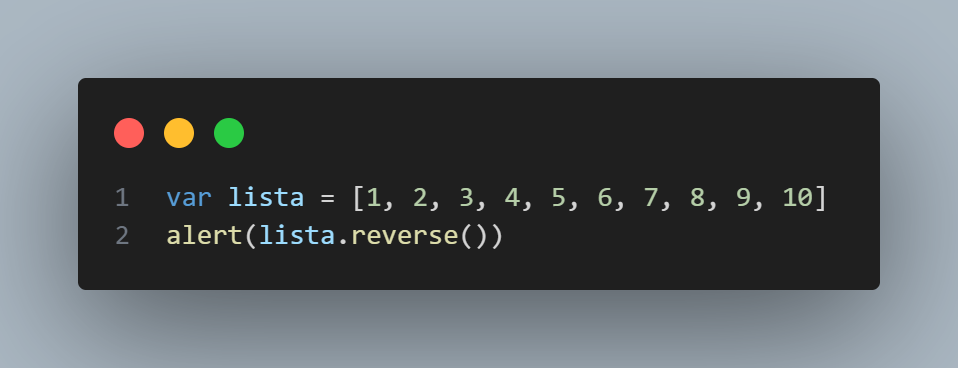
Birigui-SP

Questão 1



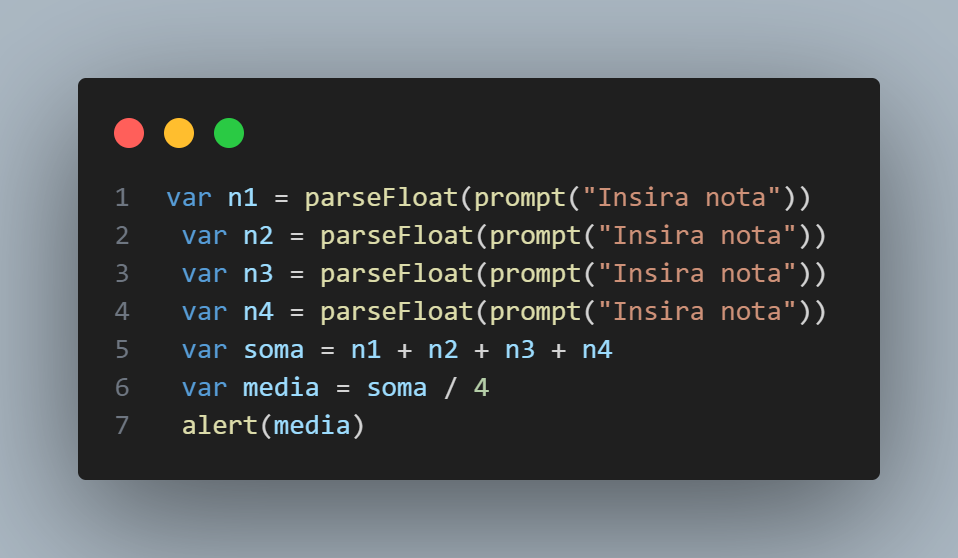
* O código começa criando uma lista (ou array) contendo os números 1, 2, 3, 4 e 5.
* O código usa um loop “for - in” para iterar sobre as propriedades do array. No caso de arrays, isso significa que o loop itera sobre os índices dos elementos (0, 1, 2, 3, 4).
* O bloco do loop “for - in” está vazio, o que significa que não há nenhuma ação sendo tomada para cada índice durante a iteração.
* Após o loop, o código tenta verificar se o valor no índice atual do array é igual a 5.
* Se a condição “if” fosse verdadeira, um alerta seria exibido informando que o número 5 está presente. Caso contrário, outro alerta indicaria que o número 5 não está presente.

Questão 2



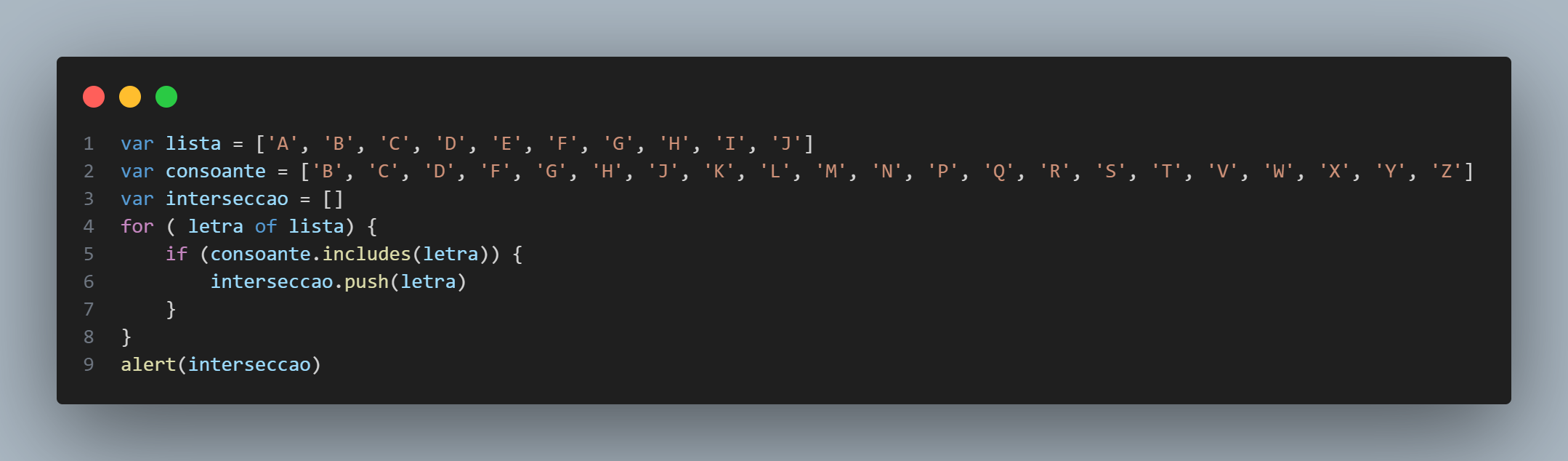
* O código começa criando uma variável chamada lista e atribui a ela um array contendo os números de 1 a 10 em ordem crescente.
* O reverse() é chamado sobre o array lista. Esse método modifica o array original, invertendo a ordem dos seus elementos. Portanto, após a chamada de reverse(), o array lista será modificado para [10, 9, 8, 7, 6, 5, 4, 3, 2, 1].
* “alert()” é usada para exibir uma mensagem de alerta na tela. Quando alert() é chamada com o array lista (após a reversão), ele irá converter o array em uma string e exibir essa string no alerta. O resultado exibido será “10, 9, 8, 7, 6, 5, 4, 3, 2, 1”.

Questão 3



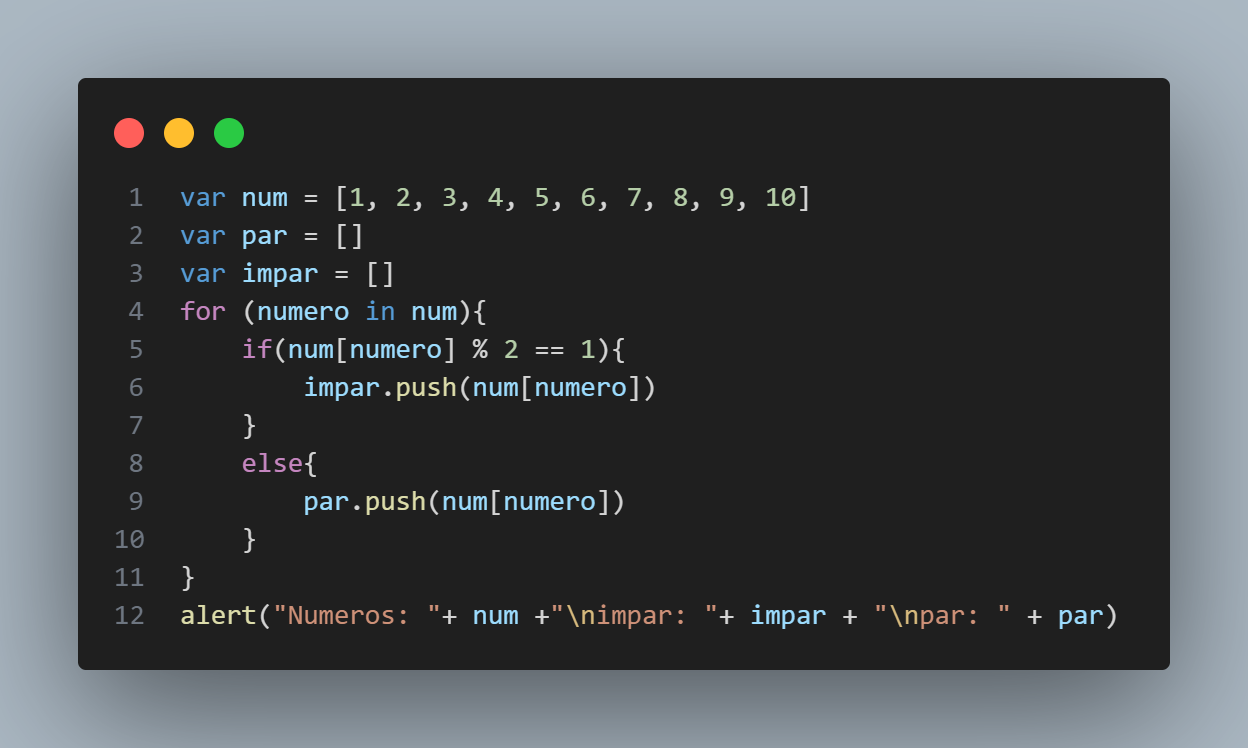
* O código solicita ao usuário que insira quatro notas através de caixas de prompt. Cada entrada é recebida como uma string.
* As strings recebidas são convertidas em números de ponto flutuante utilizando a função parseFloat.
* As quatro notas convertidas são somadas para obter a soma total.
* A soma total das notas é dividida por quatro para calcular a média.
* A média calculada é exibida em uma caixa de alerta, mostrando o resultado para o usuário.

Questão 4



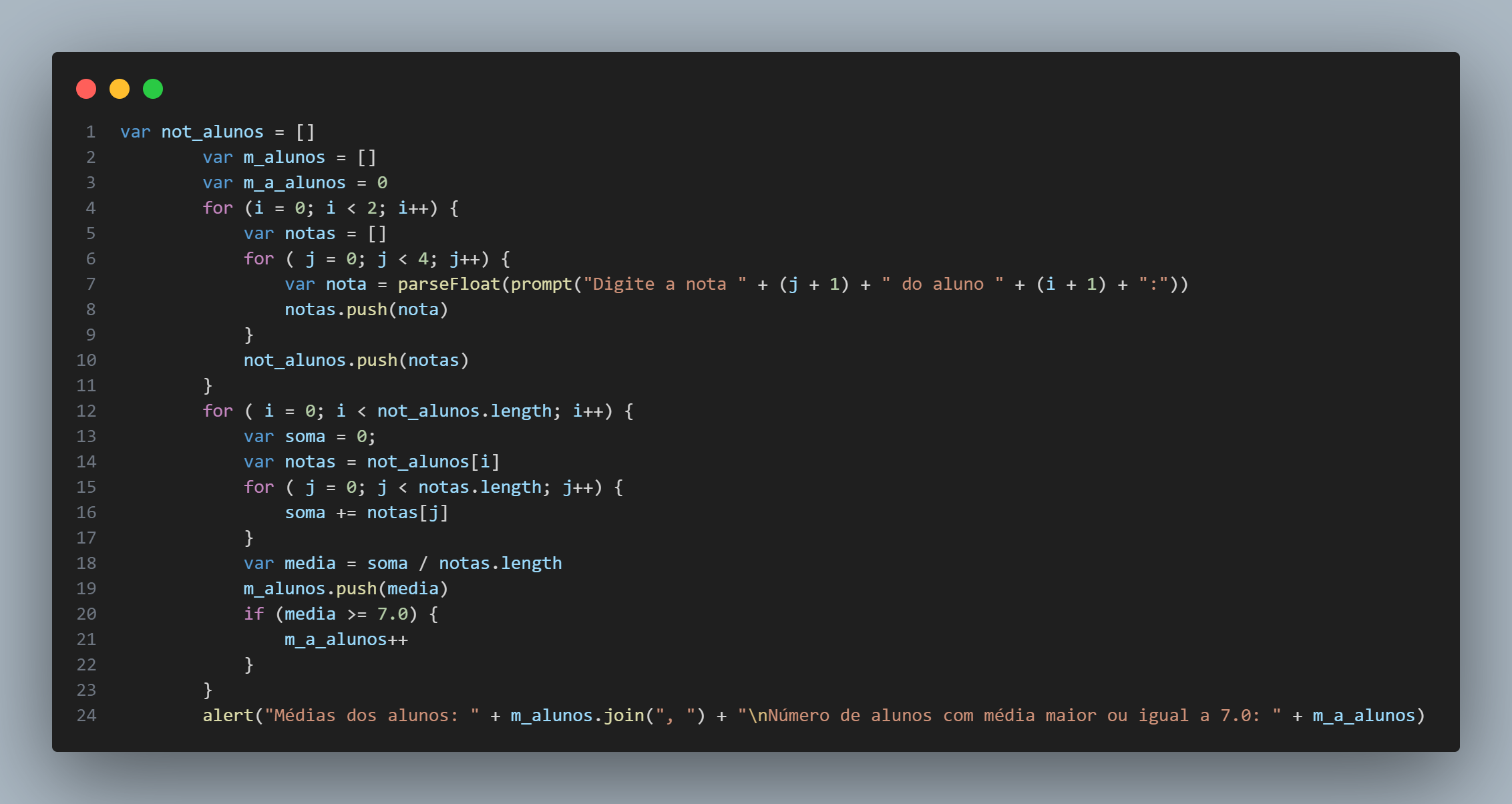
* O código começa criando duas listas: “lista”, que contém letras do alfabeto, e “consoante”, que contém letras que são consoantes.
* Um array vazio chamado “interseccao” que é feito para armazenar as letras que são encontradas em ambas as listas.
* O código usa um loop “for - of” para iterar sobre cada letra na lista “lista”.
* Para cada letra, o código verifica se essa letra está presente na lista “consoante” usando o método “includes”.
* Se a letra estiver presente em "consoante”, ela é adicionada ao array “interseccao”.
* Finalmente, o código usa a função alert() para exibir as letras que são consoantes e também estão presentes na lista “lista”. O resultado será mostrado em um alerta como uma string com as letras encontradas.

Questão 5



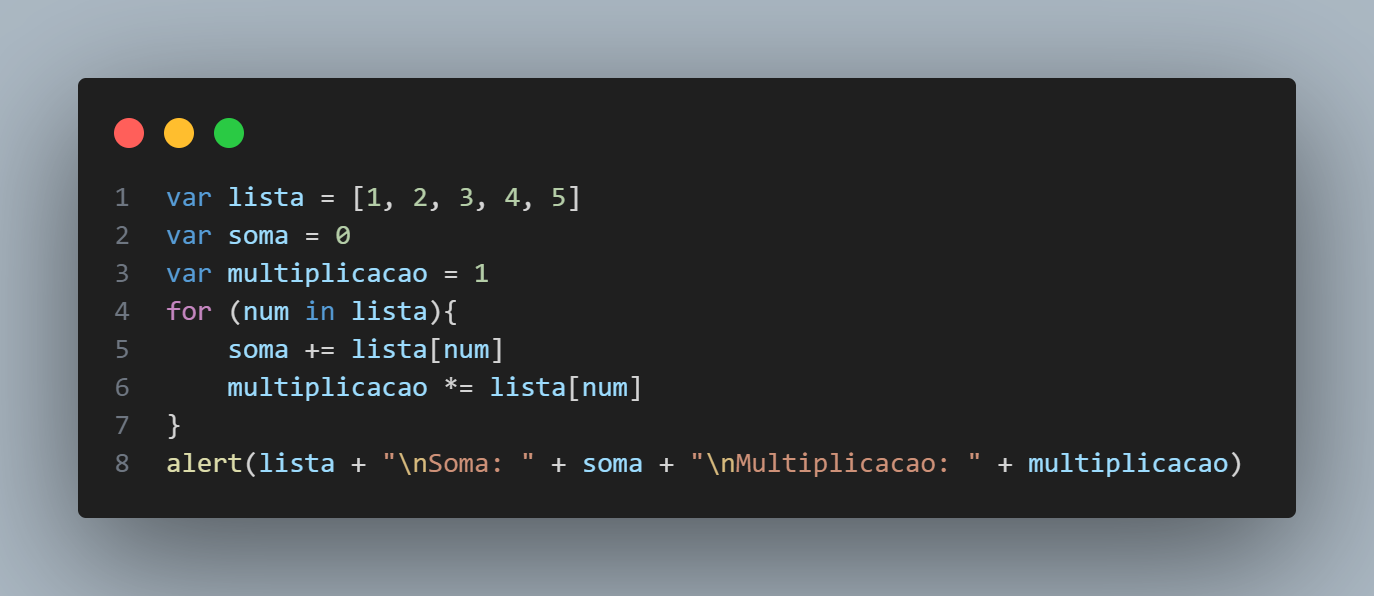
* O código começa criando um array “num” que contém os números de 1 a 10.
* Duas listas vazias, “par” e “impar”, são criadas para armazenar números pares e ímpares, respectivamente.
* O código usa um loop “for - in” para iterar sobre os índices do array “num”. O loop “for - in” fornece os índices dos elementos do array, e “numero” é a variável que armazena esses índices.
* Dentro do loop, o código verifica se o número no índice atual (num[numero]) é ímpar. Isso é feito usando o operador módulo (%). Se o número for ímpar (resto da divisão por 2 é 1), ele é adicionado ao array impar.
* Se o número for par (resto da divisão por 2 é 0), ele é adicionado ao array par.
* Finalmente, a função alert() é usada para exibir uma mensagem que mostra o array original num, os números ímpares armazenados em impar, e os números pares armazenados em par. O resultado é formatado com quebras de linha (\n) para melhor legibilidade.

Questão 6



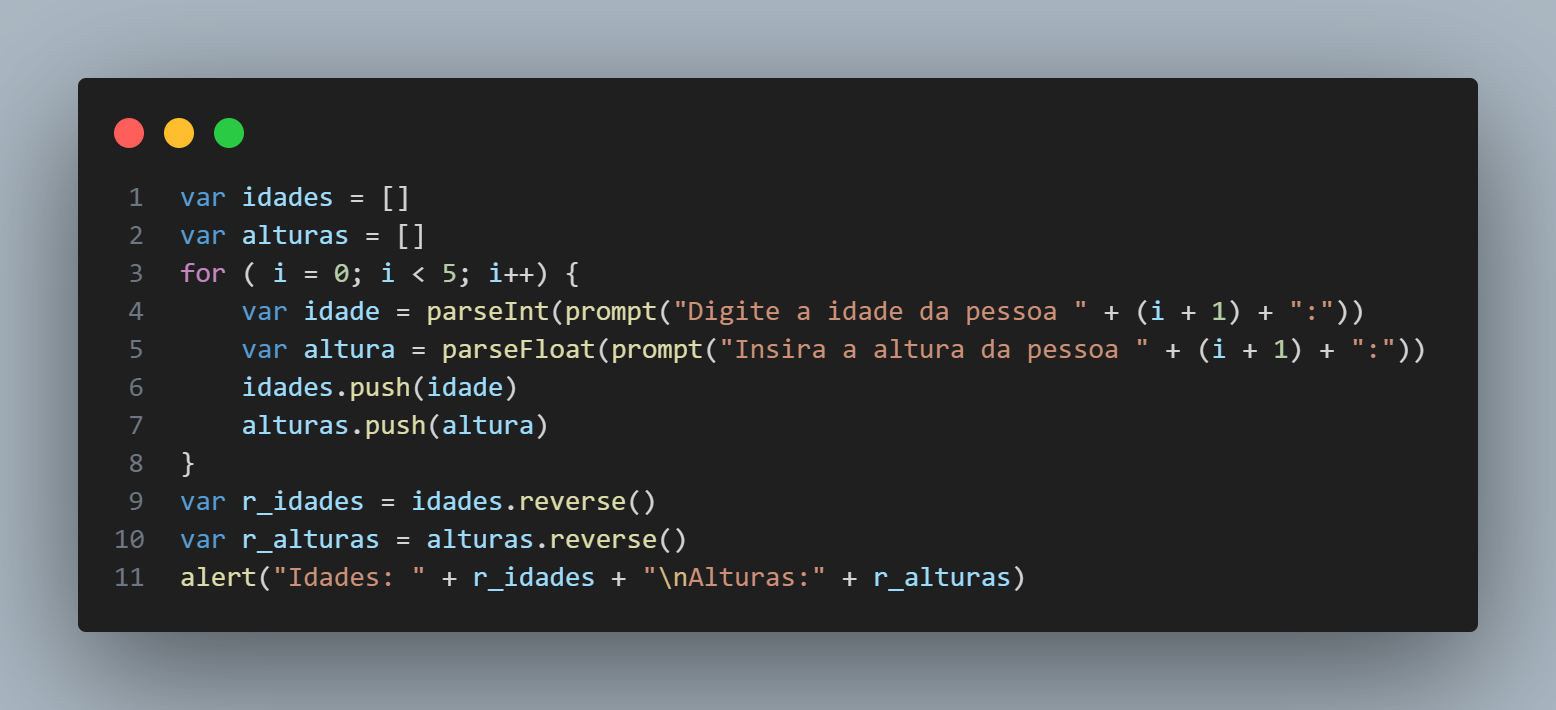
* O código começa criando três variáveis: “not\_alunos” para armazenar as notas dos alunos, “m\_alunos” para armazenar as médias dos alunos, e “m\_a\_alunos” para contar o número de alunos com média maior ou igual a 7.0.
* Um primeiro loop “for” itera duas vezes, uma para cada aluno.
* Dentro deste loop, um segundo loop “for” coleta quatro notas para cada aluno, uma de cada vez, através de prompts. As notas são convertidas para números de ponto flutuante usando “parseFloat” e são armazenadas em um array notas.
* Após coletar as quatro notas, o array notas é adicionado ao array “not\_alunos”.
* Um segundo loop for itera sobre o array “not\_alunos”, que contém as notas de todos os alunos.
* Dentro deste loop, a soma das notas de cada aluno é calculada.
* A média é então calculada dividindo a soma pelo número total de notas (4).
* A média é adicionada ao array m\_alunos.
* Se a média for maior ou igual a 7.0, o contador “m\_a\_alunos” é incrementado.
* Finalmente, a função “alert()” é usada para exibir uma mensagem que mostra as médias dos alunos (formatadas como uma lista separada por vírgulas) e o número de alunos com média maior ou igual a 7.0.
* “m\_alunos.join(", ")” transforma o array em uma string, com cada média separada por uma vírgula e um espaço. Por exemplo, se “m\_alunos” for [7.5, 8.0], o resultado de “m\_alunos.join(", ")” será "7.5, 8.0".

Questão 7



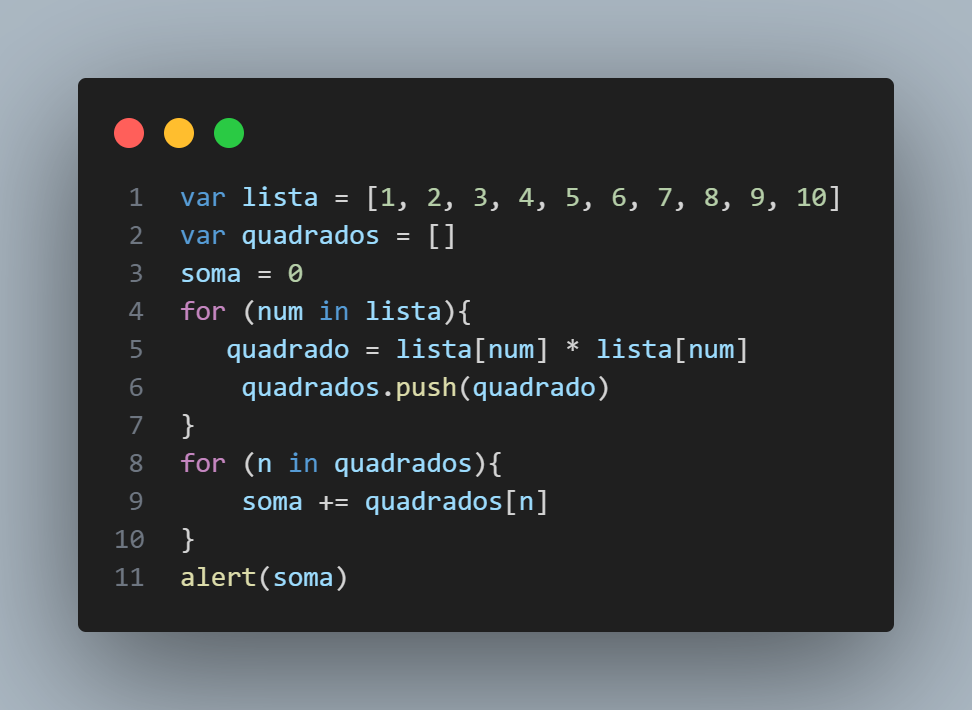
* O código começa criando um array chamado “lista” com os números de 1 a 5.
* Duas variáveis são inicializadas: “soma”, que começa em 0, e “multiplicacao”, que começa em 1.
* Um loop “for - in” é usado para iterar sobre os índices do array lista. A variável “num” representa o índice atual do array, não o valor do elemento.
* Dentro do loop, o código acessa o valor no índice “num” do array lista e o adiciona à variável “soma”. Assim, soma acumula a soma de todos os valores no array.
* O código também multiplica o valor no índice “num” pelo valor atual de “multiplicacao”. Assim, “multiplicacao” acumula o produto de todos os valores no array.
* Finalmente, a função “alert()” exibe o “array” lista, a soma dos valores e o produto dos valores. Os resultados são apresentados em uma mensagem formatada com quebras de linha “(\n)”.

Questão 8



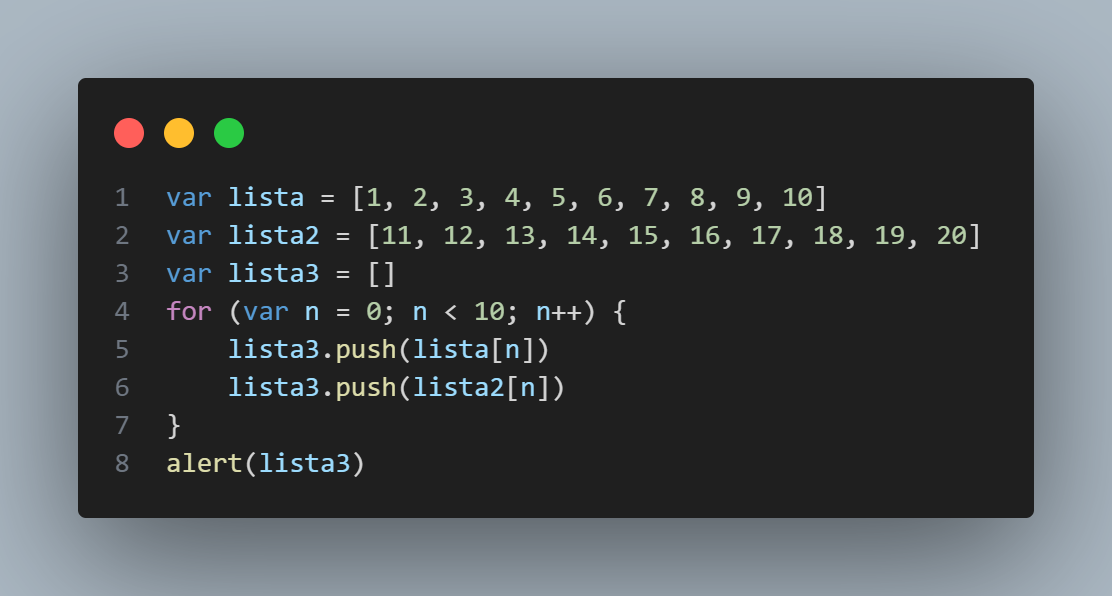
* Duas listas vazias, “idades” e “alturas”, são criadas para armazenar, respectivamente, as idades e alturas das pessoas.
* Um loop for é utilizado para iterar 5 vezes, uma vez para cada pessoa.
* Em cada iteração do loop:
* O código usa “prompt()” para solicitar a idade da pessoa e a altura, convertendo essas entradas para um número inteiro e um número de ponto flutuante, respectivamente.
* As idades e alturas são então adicionadas aos arrays idades e alturas usando o método “push()”.
* Após a coleta dos dados, o método “reverse()” é chamado em ambos os arrays (“idades” e “alturas”). Este método inverte a ordem dos elementos nos arrays.
* O array “r\_idades” armazenará as idades na ordem inversa em que foram inseridas.
* O array “r\_alturas” armazenará as alturas na ordem inversa em que foram inseridas.
* A função “alert()” é utilizada para exibir uma mensagem que mostra as idades e alturas na ordem inversa. A mensagem é formatada para mostrar as listas separadas por quebras de linha “(\n)”.

Questão 9



* O código começa criando um array “lista” com os números de 1 a 10.
* Um array vazio chamado “quadrados” é criado para armazenar os quadrados dos números.
* A variável “soma” é inicializada como 0 para acumular a soma dos quadrados.
* Um loop “for - in “é utilizado para iterar sobre os índices do array “lista”. A variável “num” representa o índice atual.
* Dentro do loop, o quadrado do número no índice “num” do array “lista” é calculado multiplicando o número por ele mesmo.
* O quadrado calculado é adicionado ao array “quadrados” usando o método “push()”.
* Outro loop “for - in” é usado para iterar sobre os índices do array quadrados. A variável “n” representa o índice atual.
* Dentro deste loop, a soma dos quadrados é calculada acumulando os valores do array quadrados na variável soma.
* Finalmente, a função “alert()” é utilizada para exibir a soma dos quadrados dos números do array “lista”.

Questão 10

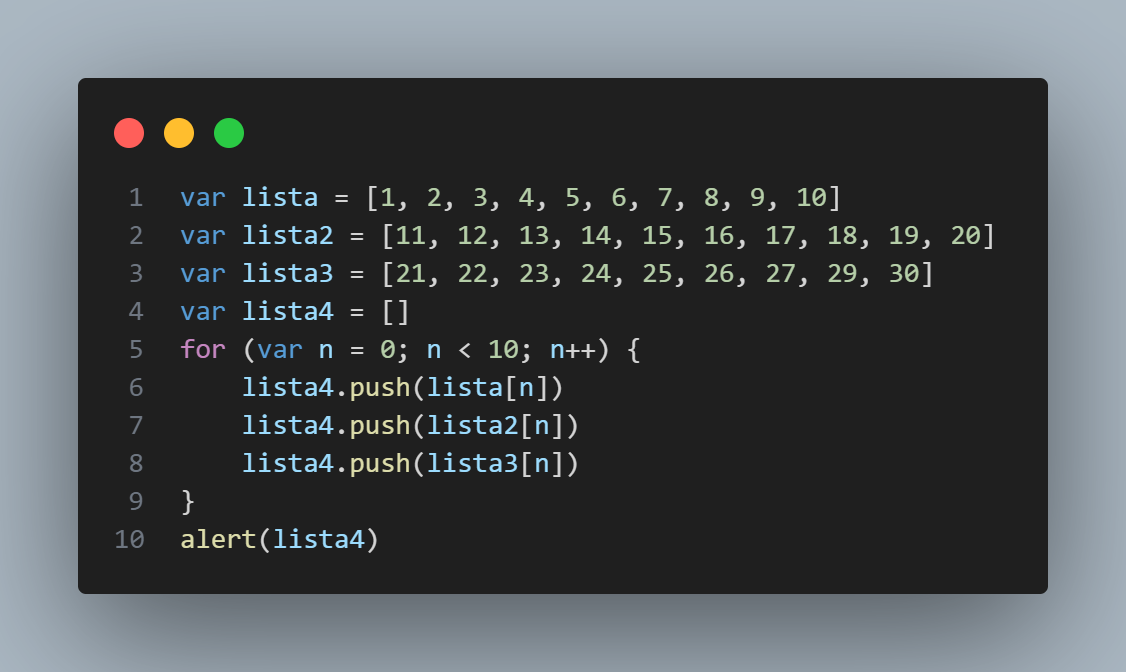


* O código começa criando dois arrays: “lista” e “lista2”. O primeiro contém números de 1 a 10 e o segundo contém números de 11 a 20.
* Um array vazio chamado “lista3” é criado para armazenar os elementos combinados de “lista” e “lista2”.
* O loop “for” é usado para iterar de 0 a 9 (10 iterações no total).

Dentro do loop:

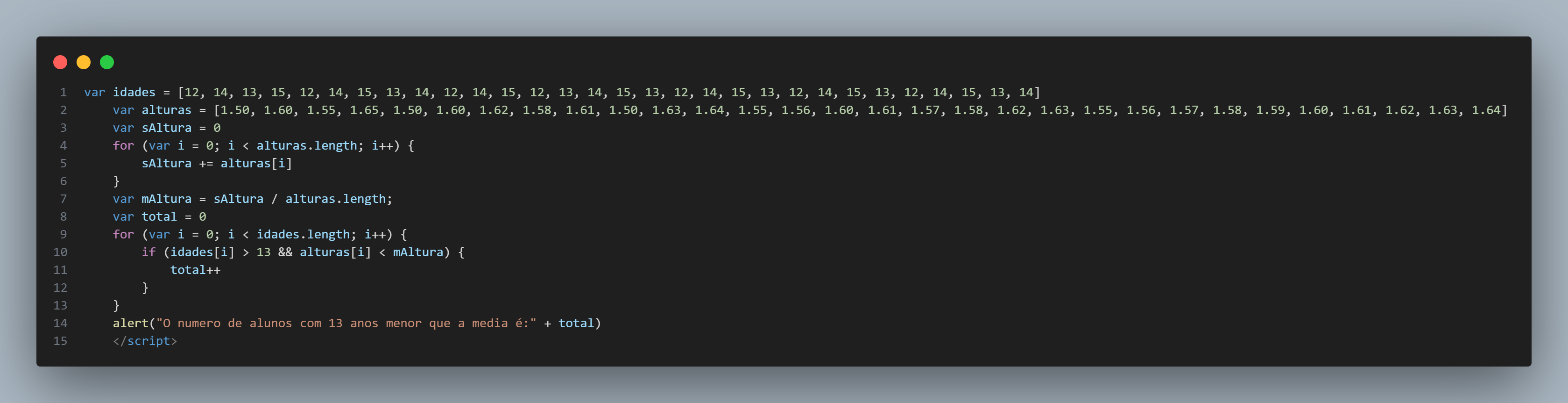
* O elemento no índice “n” do array “lista” é adicionado ao array “lista3” usando o método “push()”.
* Em seguida, o elemento no índice “n” do array “lista2” é também adicionado ao array “lista3”.
* A função “alert()” exibe o conteúdo do array “lista3”. Este array contém os elementos de ”lista” e “lista2” intercalados.

Questão 11



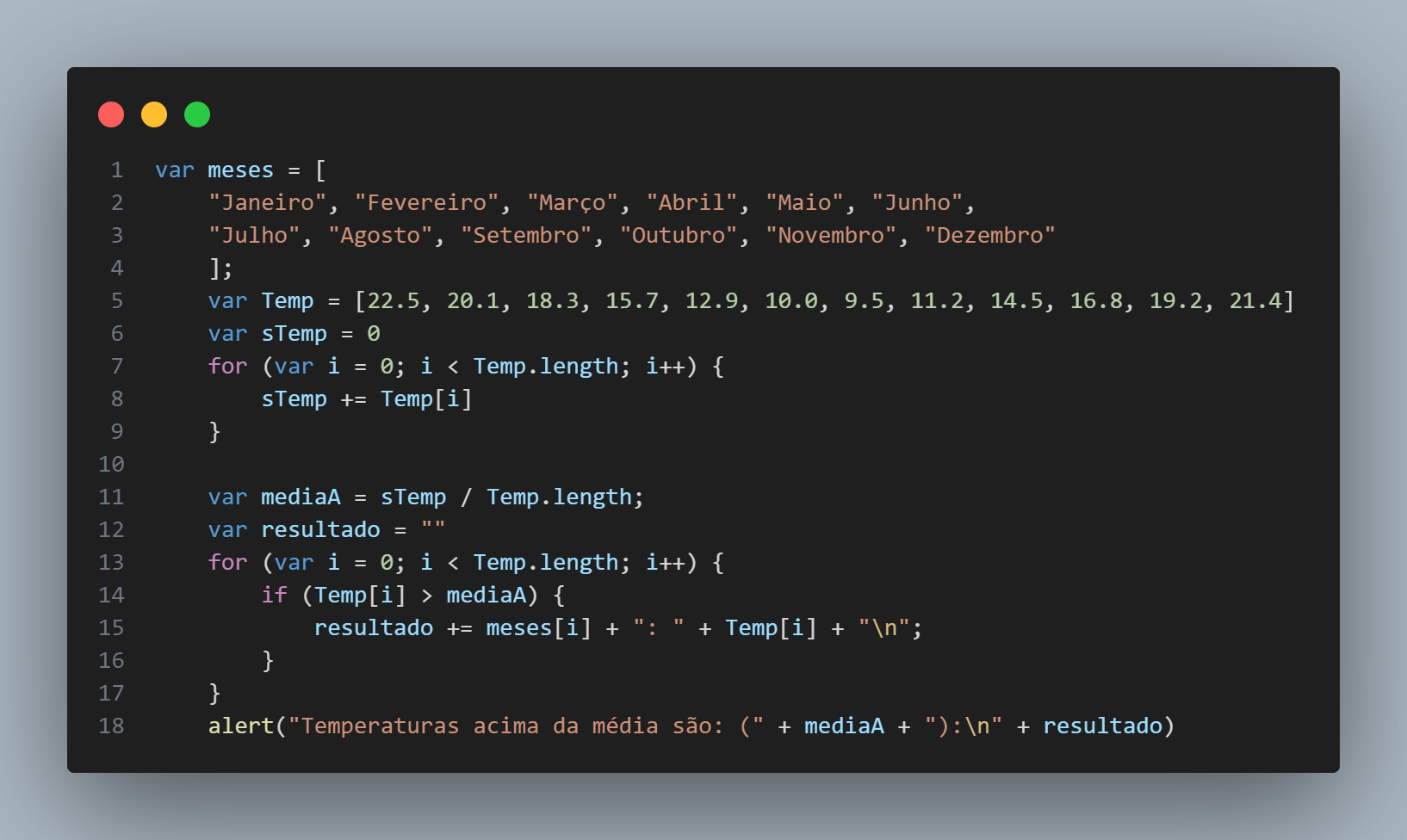
* São criados três arrays:
  + “Lista” contém os números de 1 a 10.
  + “Lista2” contém os números de 11 a 20.
  + “Lista3” contém os números de 21 a 30, mas falta o número 28.
* Um array vazio chamado Lista4 é criado para armazenar os resultados combinados.
* Um loop “for” é iniciado e vai de 0 até 9 (ou seja, 10 vezes).
* Adicionando Elementos ao Array Lista4:
* Dentro do loop, para cada número de 0 a 9:
  + O número correspondente da Lista é adicionado a Lista4.
  + O número correspondente da Lista2 é adicionado a Lista4.
  + O número correspondente da Lista3 é adicionado a Lista4.
* A Lista3 tem apenas 9 números, então quando o loop chega ao índice 9, não há um valor na Lista3 para esse índice, resultando em undefined sendo adicionado a Lista4.
* Após o loop, a função alert() exibe o conteúdo de Lista4. O array final contém os valores intercalados de Lista, Lista2 e List3.

Questão 12



* “Idades” é um array que contém as idades de vários alunos.
* “Alturas" é um array que contém as alturas correspondentes desses alunos.
* “sAltura” é uma variável que armazena a soma total das alturas. Inicialmente, é definida como 0.
* Um loop “for” percorre cada elemento do array "alturas”. Em cada iteração, o valor da altura atual “(alturas[i])” é adicionado a “sAltura”. Assim, “sAltura” acumula a soma de todas as alturas.
* Após o loop, a média das alturas é calculada dividindo “sAltura“ pelo número total de elementos no array alturas. O resultado é armazenado na variável “mAlturas”.
* “Total" é uma variável que armazena a contagem de alunos que atendem a determinados critérios. Inicialmente, é definida como 0.
* Um segundo loop “for” percorre cada elemento do array “idades”. Para cada aluno, o código verifica se a idade é maior que 13 e se a altura correspondente é menor que a média calculada “(mAltura)”.
* Se ambos os critérios forem atendidos, o contador “total” é incrementado em 1.
* Após o segundo loop, o valor de “total” (que representa o número de alunos com mais de 13 anos e altura abaixo da média) é exibido em uma caixa de diálogo usando “alert()”.

Questão 13

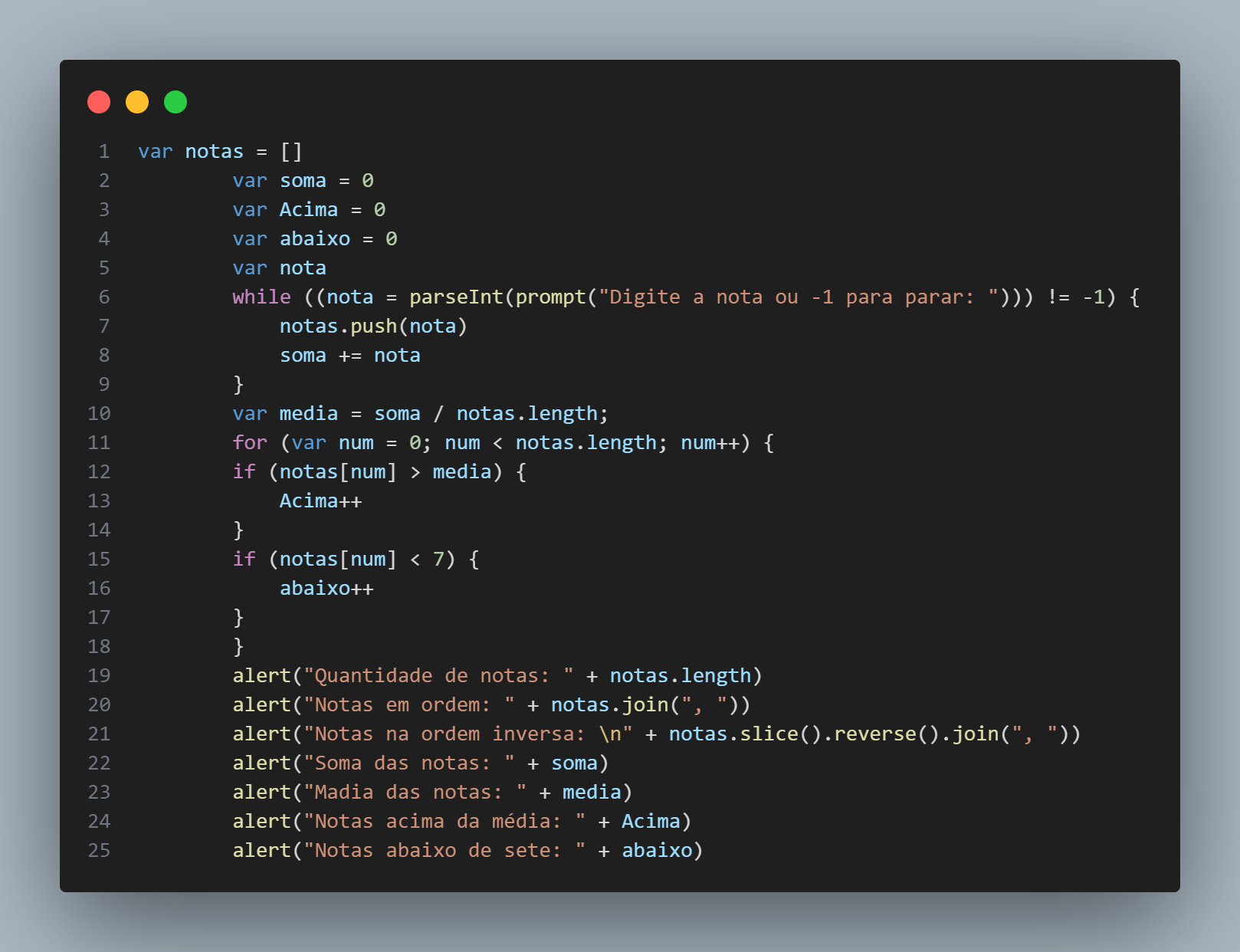
* 
* “meses” é um array que contém os nomes dos meses do ano.
* “Temp” é um array que contém as temperaturas médias para cada mês do ano.
* “sTemp" é uma variável que armazena a soma total das temperaturas. Inicialmente, é definida como 0.
* Um loop "for” percorre cada elemento do array “Temp”. Em cada iteração, o valor da temperatura atual “(Temp[i])” é adicionado a sTemp Dessa forma, sTemp acumula a soma de todas as temperaturas.
* Após o loop, a média das temperaturas é calculada dividindo sTemp pelo número total de elementos no array “Temp”. O resultado é armazenado na variável mediaA.
* resultado é uma string vazia que será usada para armazenar os meses e suas temperaturas que estão acima da média.
* Um segundo loop “for” percorre cada elemento do array “Temp”. Para cada temperatura, o código verifica se a temperatura é maior que a média calculada “(mediaA)”.
* Se a temperatura for maior que a média, o nome do mês correspondente (obtido a partir do array “(meses[i])” e a temperatura são adicionados à string “resultado”, formatados como "Mês: Temperatura" e seguidos por uma nova linha (\n).
* Após o segundo loop, a função “alert()” exibe uma mensagem que contém a média das temperaturas e a lista de meses e temperaturas que estão acima dessa média.

Questão 14



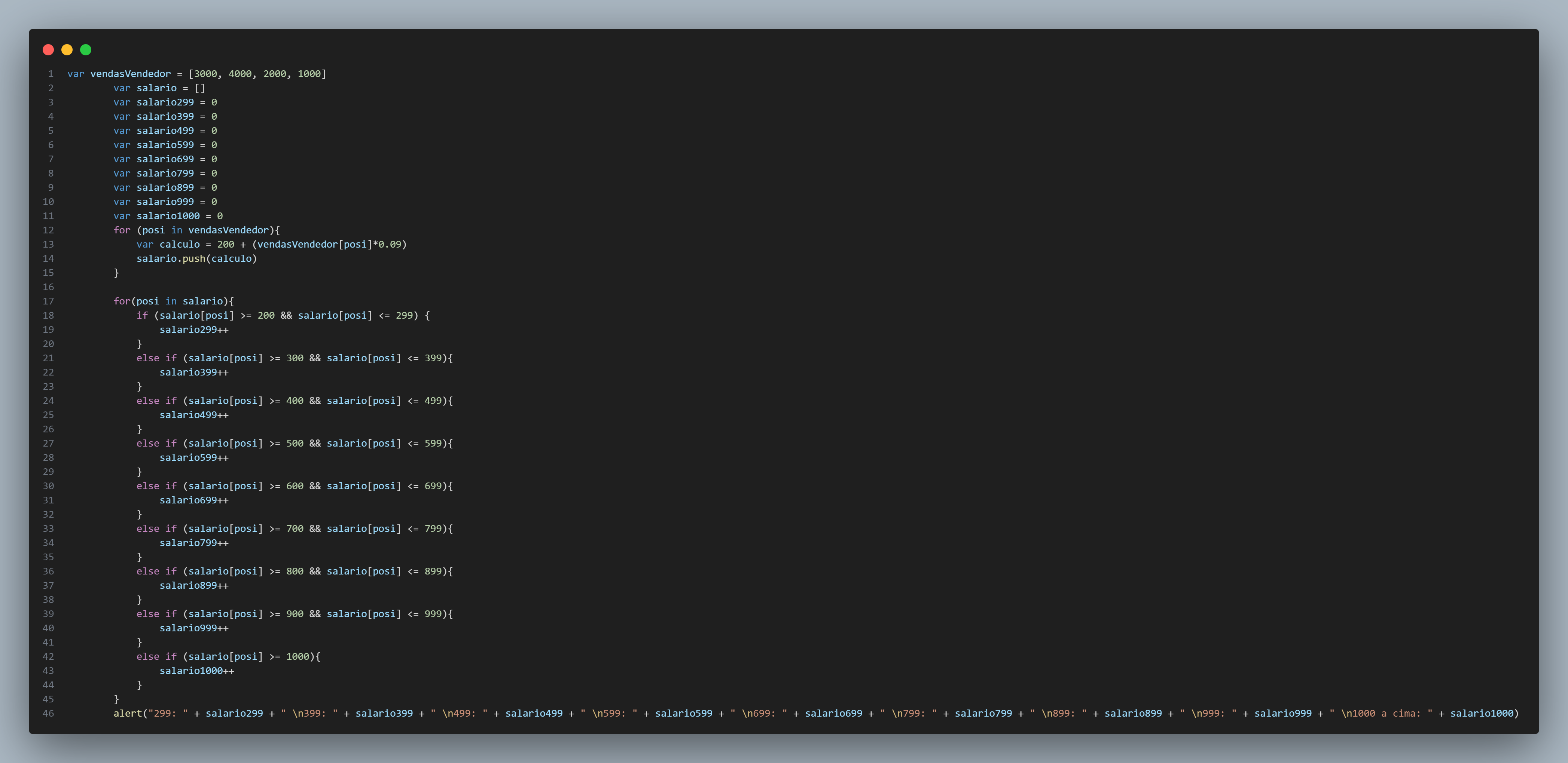
* O código usa a função “prompt()” para exibir uma série de perguntas ao usuário, solicitando respostas sobre seu envolvimento em relação a um crime. As respostas são armazenadas nas variáveis p1, p2, p3, p4, e p5.
* As perguntas são:
  + "Telefonou para a vítima?"
  + "Esteve no local do crime?"
  + "Mora perto da vítima?"
  + "Devia para a vítima?"
  + "Já trabalhou com a vítima?"
* “nivel” é uma variável que armazena a quantidade de respostas afirmativas (respostas "S"), representando o quão culpado a pessoa seria. Inicialmente, “nivel” é definido como 0.
* Para cada resposta (p1, p2, p3, p4, p5):
  + A resposta é convertida para maiúsculas com “toUpperCase()” para garantir que a comparação não seja sensível a maiúsculas e minúsculas.
  + Se a resposta for "S" (indicando uma resposta afirmativa), o contador “nivel” é incrementado em 1.
* Com base no valor final de “nivel”, o código usa uma série de “if” para determinar o grau de envolvimento:
  + Se “nivel” for 1 ou menor, a mensagem "Inocente" é exibida com a função “alert()”.
  + Se “nivel”for exatamente 2, a mensagem "Suspeito" é exibida.
  + Se “nivel”for 3 ou 4, a mensagem "Cumplice" é exibida.
  + Se “nivel” for maior que 4, a mensagem "Assasino" é exibida.

Questão 15



* “notas” é um array vazio que será usado para armazenar as notas digitadas pelo usuário.
* “soma” é uma variável que armazena a soma total das notas. Inicialmente, é definida como 0.
* “acima” é uma variável que conta quantas notas estão acima da média. Inicialmente, é definida como 0.
* “abaixo" é uma variável que conta quantas notas estão abaixo de 7. Inicialmente, é definida como 0.
* “nota" é uma variável que armazenará a nota digitada pelo usuário durante o loop.
* O código usa um loop “while” para coletar notas do usuário. O loop continua até que o usuário digite -1.
* Dentro do loop, a função “prompt()” exibe uma caixa de diálogo pedindo ao usuário para digitar uma nota ou -1 para parar.
* “parseInt()” é usado para converter a entrada do usuário de string para número inteiro.
* Se o valor digitado não for -1, a nota é adicionada ao array “notas” com “notas.push(nota)”, e a nota é somada à variável “soma”.
* Após o loop, a média das notas é calculada dividindo a soma pelo número total de notas “(notas.length)”.
* Contagem de Notas Acima da Média e Abaixo de 7:
* Um loop for percorre cada nota no array “notas”.
* Se uma nota é maior que a média, o contador “Acima” é incrementado.
* Se uma nota é menor que 7, o contador abaixo” é incrementado.
* O código usa a função “alert()” para mostrar várias informações:
  + A quantidade total de notas.
  + As notas em ordem de entrada.
  + As notas na ordem inversa.
  + A soma das notas.
  + A média das notas.
  + O número de notas acima da média.
  + O número de notas abaixo de 7.

Questão 16



* “vendasVendedor" é um array que contém valores de vendas de um vendedor.
* “salario" é um array vazio onde serão armazenados os salários calculados.
* As variáveis “salario299”, “salario399”, “salario499”, “salario599”, “salario699”, “salario799”, “salario899”, “salario999”, e “salario1000” são usadas para contar quantos salários se enquadram em diferentes faixas de valores. Todas são inicializadas como 0.
* Um loop “for...in” é usado para iterar sobre cada índice do array “vendasVendedor”.
* Para cada venda "(vendasVendedor[posi])”, o salário correspondente é calculado como “200 + (vendasVendedor[posi] \* 0.09)”. Aqui, 200 é um valor fixo e 9% da venda é adicionado ao salário base.
* O salário calculado é adicionado ao array “salario” usando o método “push()”.
* Um segundo loop “for...in” é usado para iterar sobre cada índice do array “salario”.
* Para cada salário “(salario[posi])”, o código verifica em qual faixa de valores ele se encaixa e incrementa o contador correspondente:
  + Se o salário está entre 200 e 299, o contador “salario299” é incrementado.
  + Se o salário está entre 300 e 399, o contador “salario399” é incrementado.
  + E assim por diante, até a faixa de 1000 e acima “(salario1000)”.
* A função “alert()” exibe uma mensagem que lista a quantidade de salários em cada faixa. A mensagem é formatada para mostrar quantos salários estão em cada intervalo de valores.

Questão 17



* O código usa a função “prompt()” para solicitar ao usuário que digite o nome do atleta. O valor inserido é armazenado na variável nome.
* O código usa “prompt()” para solicitar cinco medidas de saltos, uma para cada salto.
* Cada valor digitado é convertido para um número de ponto flutuante usando “parseFloat()”, e os valores são armazenados nas variáveis “salto1”, “salto2”, “salto3”, “salto4”, e “salto5”.
* A média dos cinco saltos é calculada somando todos os saltos e dividindo o total por 5. O resultado é armazenado na variável “media”.
* A função “alert()” é usada para exibir várias mensagens:
  + Primeiro, exibe o nome do atleta.
  + Em seguida, exibe a média dos saltos.
  + Por último, exibe os valores individuais dos saltos, cada um em uma linha separada.

Questão 18



* “votos” é um array que armazenará os votos dos usuários.
* “t\_votos” é uma variável que conta o total de votos inseridos.
* “mlrJogador” armazena o número da camisa do jogador com mais votos.
* “MlrJogadorVotos" armazena o número de votos do jogador com mais votos.
* “mlrJogadorPerc” armazena o percentual de votos do jogador com mais votos.
* Um loop “while (true)” é usado para continuar coletando votos até que o usuário decida parar.
* O código solicita que o usuário insira o número da camisa do jogador ou digite 0 para encerrar.
* Se o valor inserido for 0, o loop “while” é interrompido usando “break”.
* Se o valor inserido estiver fora do intervalo de 1 a 23, é exibida uma mensagem de erro e o loop continua pedindo a entrada.
* Se o valor for válido, ele é adicionado ao array “votos”, e o contador “t\_votos” é incrementado.
* Um loop “for” percorre todos os votos no array votos.
* Para cada jogador (número da camisa) no array “votos”, um contador “VtJogador” é usado para contar quantos votos esse jogador recebeu.
* Um segundo loop “for” conta as ocorrências do jogador atual no array “votos”.
* O percentual de votos para o jogador é calculado como “(VtJogador / t\_votos) \* 100”.
* Se o jogador atual tiver mais votos do que o jogador com mais votos registrado “(mlrJogadorVotos)”, atualiza as variaveis “mlrJogador”, “mlrJogadorVotos” e “mlrJogadorPerc” com os valores do jogador atual.
* Exibe o total de votos computados.
* Outro loop for percorre o array “votos” novamente para exibir o número de votos e o percentual para cada jogador.
* No final, exibe o número do jogador com mais votos, o total de votos que ele recebeu, e o percentual de votos.

Questão 19



* “c1" a “c6” são contadores que armazenarão o número de votos para cada uma das opções disponíveis:
  + “c1" para "Windows Server"
  + “c2” para "Unix"
  + “c3” para "Linux"
  + “c4” para "Netware"
  + “c5” para "Mac OS"
  + “c6” para "Outro"
* “eleitor" é um contador que registra o total de votos recebidos.
* Um loop while (true) é usado para continuar coletando votos até que o usuário decida sair.
* Dentro do loop, a função “prompt()” exibe um menu para o usuário escolher uma das opções:
  + 1-Windows Server
  + 2-Unix
  + 3-Linux
  + 4-Netware
  + 5-Mac OS
  + 6-Outro
  + 0-Sair para encerrar a votação.
* A entrada do usuário é convertida para um número inteiro usando “parseInt()” e armazenada na variável voto.
* O código verifica se a entrada não é um número “(isNaN(voto))” (isNaN: “Not a Number” ou seja, verifica se é um numero ou naõ), ou se está fora do intervalo de 0 a 6. Se for o caso, exibe uma mensagem de erro e continua solicitando uma entrada válida.
* Se o voto for 0, o loop é interrompido com “break”, encerrando a coleta de votos.
* Se o voto estiver entre 1 e 6, o contador correspondente é incrementado:
  + Se “voto” é 1, incrementa “c1”.
  + Se “voto” é 2, incrementa “c2”, e assim por diante.
* Após sair do loop, a função “alert()” exibe os resultados da votação.
* Para cada opção, o percentual de votos é calculado como “(contagem de votos / total de eleitores) \* 100” e formatado com duas casas decimais usando “toFixed(2)”.
* O total de eleitores é exibido no final.

Questão 20

* 
* “salarios” é um array que armazenará os salários informados.
* “abonos” é um array que armazenará os abonos calculados para cada salário.
* “totalG"é a variável para acumular o total gasto com abonos.
* “abonoMin” é o valor mínimo para o abono, fixado em R$ 100.
* “funcionariosMin” conta o número de funcionários que recebem o abono mínimo.
* “maiAbono” armazena o maior valor de abono pago.
* Um loop “while (true)” é usado para continuar coletando salários até que o usuário decida encerrar.
* Dentro do loop, a função “prompt()” solicita ao usuário que informe o salário ou digite 0 para encerrar. A entrada é convertida para um número de ponto flutuante usando “parseFloat()”.
* Se o valor do salário for 0, o loop é interrompido com “break”.
* O salário é adicionado ao array “salarios”.
* O abono é calculado como 20% do salário. Se o abono calculado for menor que R$ 100, o abono é ajustado para o valor mínimo de R$ 100 e o contador “funcionariosMin” é incrementado.
* O abono é adicionado ao array “abonos”.
* O total gasto com abonos “(totalGasto)” é incrementado pelo valor do abono.
* Se o abono calculado for maior que o maior valor registrado “(maiAbono)”, o “maiAbono” é atualizado.
* Após sair do loop, a variável resultado é inicializada para armazenar uma “string” formatada com os salários e abonos.
* Um loop “for” percorre todos os salários e abonos, formatando e adicionando a string “resultado”.
* A função “alert()” exibe os resultados:
  + Primeiro, exibe a lista de salários e abonos.
  + Em seguida, informa quantos colaboradores foram processados.
  + Exibe o total gasto com abonos.
  + Informa quantos colaboradores receberam o valor mínimo de abono.
  + Exibe o maior valor de abono pago.

Questão 21

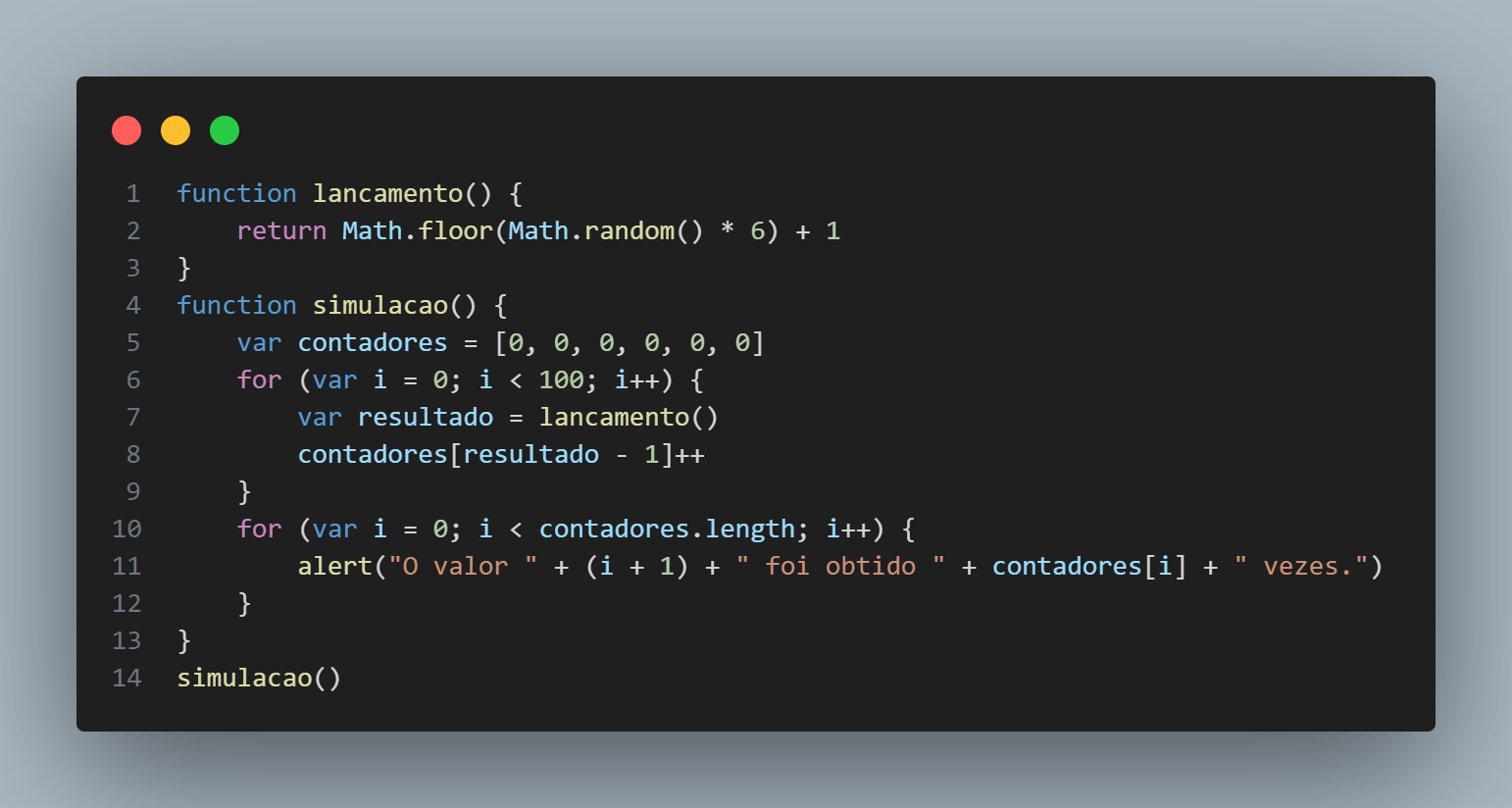
* 
* “modelos" é um array que contém os nomes de diferentes modelos de carros.
* “consumos” é um array que contém o consumo de gasolina (em km por litro) para cada modelo de carro correspondente.
* “gasolina’ é o preço do litro de gasolina, definido como R$ 2,25.
* “economico” é uma variável que irá armazenar o índice do carro com o menor consumo de gasolina.
* “relatorio” é uma string que será usada para acumular as informações de consumo e custo de cada modelo de carro.
* Um loop “for” percorre todos os índices dos arrays modelos e consumos.
* Dentro do loop:
  + “litros” calcula quantos litros de gasolina são necessários para percorrer 1000 km, com base no consumo do carro. O resultado é arredondado para uma casa decimal usando “toFixed(1)”.
  + “custo” calcula o custo total da gasolina necessária para percorrer 1000 km, multiplicando o número de litros pelo preço da gasolina. O resultado é arredondado para duas casas decimais usando “toFixed(2)”.
  + A string “relatorio” é atualizada com a informação do modelo, consumo, litros necessários e custo total.
  + O código verifica se o consumo do carro atual é menor do que o consumo do carro considerado mais econômico “(consumos[economico])”. Se for, o índice “economico” é atualizado para o índice atual.
* Após o loop, a função “alert()” exibe o relatório completo, que inclui informações sobre cada modelo de carro e seu custo para percorrer 1000 km.
* Em seguida, o código exibe qual modelo de carro é o mais econômico.

Questão 22

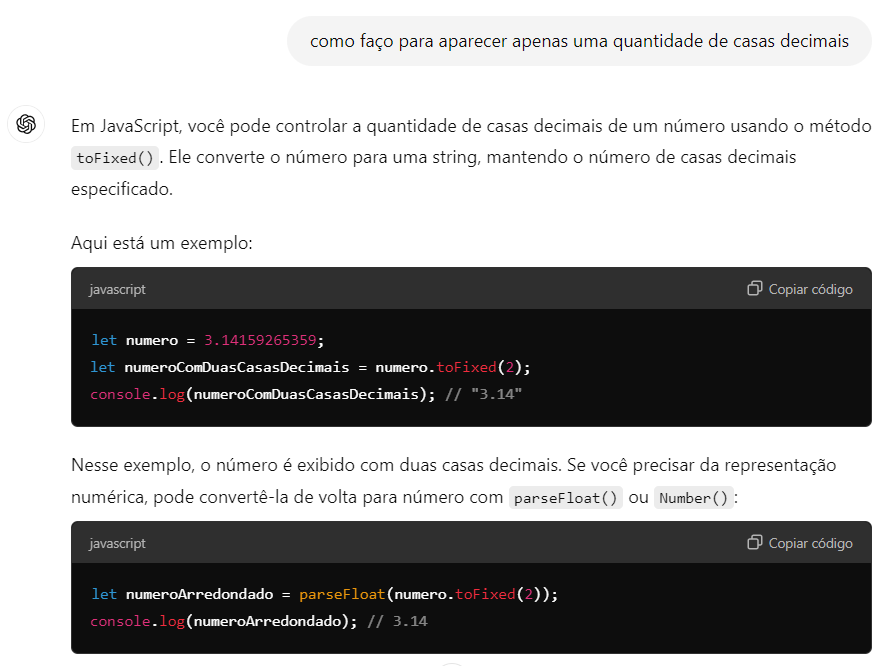
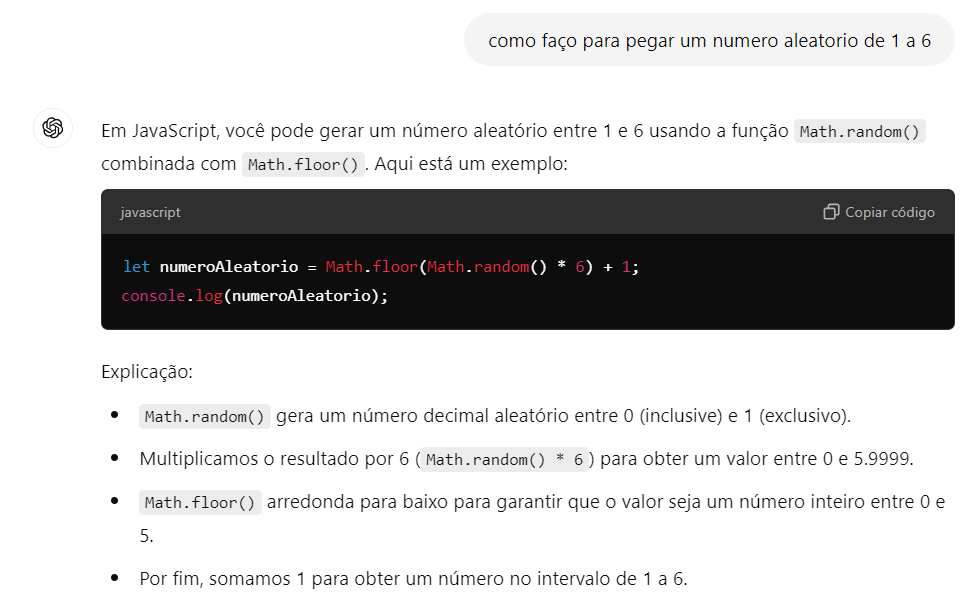
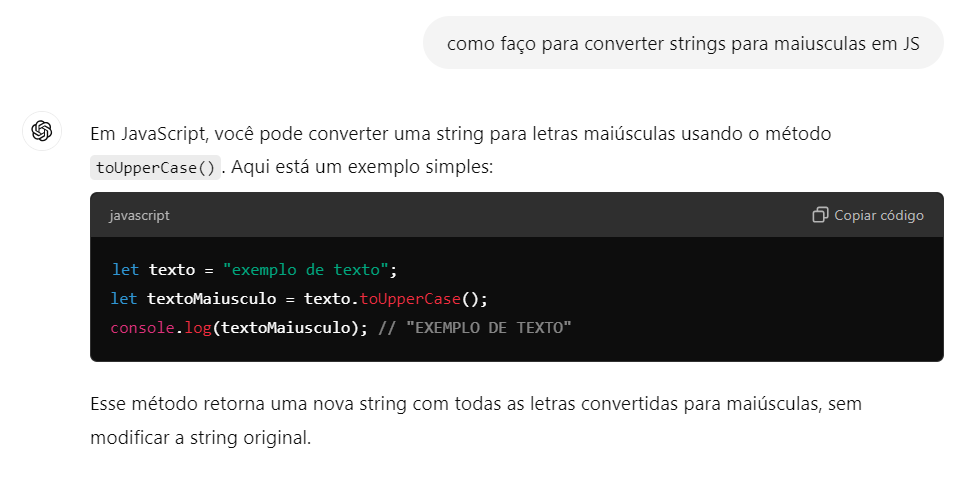


* “esfera”, “limpeza”, “cabo”, e “quebrado” são variáveis para contar a quantidade de mouses que precisam de cada tipo de manutenção:
  + esfera para contar mouses que precisam de esfera.
  + limpeza para contar mouses que precisam de limpeza.
  + cabo para contar mouses que precisam de troca de cabo ou conector.
  + quebrado para contar mouses quebrados ou inutilizados.
* “quantidade” armazena o número total de mouses, que é solicitado ao usuário.
* “mouses” é um array que armazenará objetos, cada um representando um mouse e suas necessidades de manutenção.
* Um loop for é usado para iterar a partir de 0 até quantidade - 1, onde “i” é o índice do loop.
* Para cada mouse:
  + Um objeto mouse é criado para armazenar informações sobre o mouse atual.
  + O usuário é solicitado a informar se o mouse precisa de esfera, limpeza, troca de cabo ou se está quebrado, usando “prompt()”. Os valores são armazenados nas propriedades correspondentes do objeto mouse.
  + As contagens “(esfera, limpeza, cabo, quebrado)” são incrementadas conforme as respostas do usuário.
  + O objeto “mouse” é adicionado ao array “mouses”.
* As porcentagens são calculadas para cada tipo de necessidade de manutenção:
  + “esferaPcT”, “QuebradoPcT", “limpezaPcT”, e “caboPcT” são calculados como a proporção de mouses que precisam de cada tipo de manutenção, multiplicada por 100 para obter a porcentagem.
* A função “alert()” é usada para mostrar:
  + O total de mouses.
  + A porcentagem de mouses que precisam de esfera.
  + A porcentagem de mouses que precisam de limpeza.
  + A porcentagem de mouses que precisam de troca de cabo ou conector.
  + A porcentagem de mouses que estão quebrados ou inutilizados (com o cálculo incorreto que deve ser ajustado).

Questão 24

* 
* Função “lancamento”:
* Objetivo: Simular o lançamento de um dado de seis lados.
* Como funciona:
  + “Math.random()” gera um número decimal aleatório entre 0 (inclusive) e 1 (exclusivo).
  + “Math.random() \* 6” multiplica esse número por 6, resultando em um número decimal entre 0 (inclusive) e 6 (exclusivo).
  + “Math.floor(...)” arredonda esse número para baixo, resultando em um número inteiro entre 0 e 5.
  + “Math.floor(Math.random() \* 6) + 1” ajusta o intervalo para 1 a 6 (inclusive), que são os possíveis resultados do dado.
* Retorno: Um número inteiro aleatório entre 1 e 6, simulando o resultado de um dado.
* Função “simulacao”:
* Objetivo: Simular o lançamento de um dado 100 vezes e contar quantas vezes cada valor (de 1 a 6) aparece.
* Como funciona:
  + “var contadores = [0, 0, 0, 0, 0, 0]” cria um array com 6 elementos, todos inicializados como 0. Cada posição do array corresponderá à contagem de um valor específico do dado (1 a 6).
  + O loop “for” executa 100 vezes (para simular 100 lançamentos do dado):
    - “var resultado = lancamento()” chama a função lancamento para obter um valor aleatório entre 1 e 6.
    - “contadores[resultado - 1]++” incrementa o contador correspondente ao valor obtido. Por exemplo, se resultado for 3, “contadores[2] (já que 3 - 1 = 2)” será incrementado em 1.
  + Outro loop “for” percorre o array “contadores” e imprime quantas vezes cada valor foi obtido:
    - “alert("O valor " + (i + 1) + " foi obtido " + contadores[i] + " vezes.")” exibe o valor do dado e a quantidade de vezes que ele apareceu.
* “simulacao()” chama a função “simulacao” para iniciar o processo de simulação.

Fontes



# Ajudas:

Dimitri e Kenzo