Fase 1

Questão 1 | resposta: b) 5
É adicionado um elemento a lista aumentando o numero de elementos de 4 para 5

Questão 2 | resposta: c) ['maçã', 'banana']
Remove o ultimo elemento "uva" da lista

Questão 3 | resposta: A) [20, 30, 40]
shift remove o primeiro elemento da lista então o "10" é removido

Questão 4 | resposta: b) [0, 1, 2, 3]
unshift (0) adiciona "0" ao inicio da lista

Questão 5 | resposta: b) [6, 7]
o array original é copiado apartir do indice 1

Questão 6 | resposta: b) [1, 4]
é removido 2 elemento na lista apartir do indice 1

Questão 7 | resposta: c) 2
É retornado o indice da primeira ocorrencia de "c"

Questão 8 | resposta: b) [2, 4, 6]
é criado um novo array com map com os elementos do array original dividos por 2

Questão 9 | resposta: c) [9, 12]
filter retorna um novo array com elementos maiores que 6

Questão 10 | resposta: d) 12
Aqui o reduce soma todos o valores do array sendo impresso 12

questão 11 | resposta a) true
include() no codigo verifica a existencia do elemento no array
questão 12 | resposta: c) "a-b-c"
concatena os elemento usando o hifen
Questão 13 | resposta: c) [1,2,3,4,5]
combina os array em um novo
questão 14 | resposta: b) ['z','y','x']
é impresso ordem inversa dos elementos
Questão 15 | resposta: c) 3
será impresso o primeiro elemento maior que 2
~~questão~~

Fase 2

Questão 16 | resposta: ~~O array é mapeado para~~
O array de impares [1,3,5] é mapeado para [3,9,15] e em seguida, somado com um inicial 10 dando 37
Questão 17 | resposta:

```
const arr = [10,15,22,34,45,60];
let processado = 0
for (let i = 0; i < arr.length; i++){
    const n = arr[i];
    if (n % 3 ===0){
        processado += n / 2;
    }
}
```

Questão 18 | resposta: o motivo é porque o splice() é mutavel, ele altera diretamente o array sobre o qual é chamado, removendo os elementos 8 e 12 dele.
As duas formas para evitar isso é:
FORMA 1: usando slice().

```
const listaOriginal = [4, 8, 12, 16];
const r = listaOriginal.slice(1,3);
console.log(listaOriginal)
console.log(r);
```

FORMA 2: usando filter()

```
const lista = [4, 8, 12, 16];
const r = lista.filter((e, ind) => ind >= 1 &&
ind <= 2);
console.log(lista)
console.log(r).
```

Questão 19 | resposta:

```
const pessoas = [
 { nome: 'Ana', cidade: 'SP' },
 { nome: 'Lucas', cidade: 'RJ' },
 { nome: 'Bruna', cidade: 'SP' },
 { nome: 'Caio', cidade: 'MG' }];
const pePorCity = pessoas.reduce((acc, p)
=> { if (!acc[p.cidade]) {
        acc[p.cidade] = [];
      }
    acc[p.cidade].push(p);
    return acc;
  }, {});
```

Questão 2o | resposta:
~~const v = [5, 8, 12, 8, 3];~~
~~const~~
find -> retorna o primeiro encotrado
filter -> retorna todos os em um array
some -> retorna um true ou false se 8 existe

Fase 3
Questão 1 | resposta: a)
no codigo (...resto) coleta o que sobrou em um novo objeto, por isso a resposta é a)
Questão 2 | resposta: a)
A desestruturação pula o indice 1 e pega o resto do array
Questão 3 | resposta: a)
a desestruturação aninhado extrai o valor de cidade de dentro de endereço
Questão 4 | resposta: b)
o espalhamento combina objeto e a propriedade "b" é sobrescrita pelo ultimo objeto
Questão 5 | resposta: e)
o console.log(x+y) está fora do escopo da função as variaveis x e y não existem globalmente
Questão 6 | resposta: c)
o espalhamento desempacota arr1 e arr2 em arr3 criando um novo array plano
Questão 7 | resposta: a)
a desestruturação tema vinou modo e opcoa. tente acessa o valor ~~restante~~ restante

questão 8 | resposta: b)
A função pega 10 e 20 como 'a' e 'b'
o resto vira [30, 40] e retorna 32
questão 9 | resposta: a)
a desestruturação aninhada em array.b1
pega o primeiro, ignora o segundo e o
b3 pega o ~~terc~~ terceiro
questão 10 | resposta: b)
o espalhamento faz uma copia. O objeto
compartilha a mesma referencia de
memoria, por isso mudança de um
afeta o outro

Fase 4
questão 11 | resposta: b)
o codigo pega o numero 4 do segundo ele-
mento da matriz e o resto que é
5, 6 e imprime 4 [[5, 6]]
questão 12 | resposta: a)
o codigo pega o valor "escuro" de
tema e o resto vira {nome: "Ana"}
por isso a resposta é "a"
questão 13 | resposta: a)
o valor de c no objeto original conti-
nua 3, mas o objeto resto vira 99
por isso o resultado é 399
questão 14 | resposta: a)
o array [1, 2, 3] é desestruturado, em-
tão x = 1 e resto = [2, 3]

o valor padrão de y vira [...nasto, x], ou seja, [2, 3, 1] por isso é impresso [2, 3, 1]

Questão 15 | resposta: C)

o contig2 tem uma copia separada de opcoes. Ao mudar o zoom nele, o contig1 não muda, por isso é impresso 100, 200