

*curso superior de tecnologia em
análise e desenvolvimento de sistemas*

*eletiva de programação I
programação script*

prof. Carlos Eduardo Câmara (dinho)

Aula 15: funções. Exercícios ...

TADS – JavaScript

função: exercício:

Construir um simulador de um jogo de moedas onde o usuário deve escolher a quantidade de moedas em jogo, 2, 3, 4 ou 5 moedas. Pode ser permitido ao jogador, entrar com o número de jogadas máximo que ele arrisca para o simulador acertar! Cara da moeda poder igual a zero (0) e Coroa igual a um (1).

o jogo deve gerar sequências de jogadas aleatórias do tipo Cara, Coroa (todas as possibilidades com 2 moedas); Cara, Coroa, Coroa (todas as possibilidades com 3 moedas) ; Cara, Cara, Coroa, Cara (todas as possibilidades com 4 moedas), a depender da escolha do jogador (numero de moedas entrada via console)

Deve ser gerada uma sequência de moedas inicial, que é a jogada inicial do simulador.

construir um simulador do jogo de azar de lançamento de moedas, onde devem ser geradas prováveis jogadas e comparadas com a sequência inicialmente gerada e as jogadas devem ser armazenadas em um vetor de jogadas. Quando o simulador acertar a jogada gerada inicialmente, deve ser apresentado, via console, a jogada inicialmente gerada, quantas jogadas o simulador realizou até acertar e quais foram as sequências dessas tentativas até o acerto.

Informar o jogador se acertou seu palpite.

Usar apenas o material visto em sala de aula ou da Cisco Academy – evitar outros apoios.

TADS – JavaScript

bibliografia

SILVA, M. S., JavaScript Essencial, Novatec, 2022

MORRISON, M., Use a Cabeça JavaScript, ed O'Reilly, Alta Books, 2009

ACKERMAN, P., JavaScript the comprehensive guide, Ed. Rheinwerk, 2022

W3C- resource: <https://www.w3resource.com/javascript/javascript.php> - 20/05/2024

CISCOACADEMY - <https://www.netacad.com/>

(<https://www.netacad.com/launch?id=062c7386-e936-452f-a584-65e0a3355ced&tab=curriculum&view=9ed3b581-161c-5b13-84c8-2074e7584032>)

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript> _ Site da Mozilla Developers

Alguns exercícios de livros de C

Exercício :

joga moedas

$\text{Math.random()} \leq 0.5 \rightarrow \text{Cara}$, caso contrário é Coroa

PROBLEMAS 1 SAÍDA TERMINAL E

Info: Start process (2:20:59 AM)

jogada 1: Coroa,Cara,Cara

jogada 2: Coroa,Coroa,Coroa

jogada 3: Coroa,Coroa,Coroa

jogada 4: Cara,Coroa,Cara

jogada 5: Coroa,Coroa,Coroa

jogada 6: Coroa,Coroa,Coroa

jogada 7: Cara,Cara,Cara

jogada 8: Coroa,Coroa,Coroa

jogada 9: Coroa,Coroa,Coroa

jogada 10: Cara,Cara,Cara

historico completo: [

```
[ 'Coroa', 'Cara', 'Cara' ],  
[ 'Coroa', 'Coroa', 'Coroa' ],  
[ 'Coroa', 'Coroa', 'Coroa' ],  
[ 'Cara', 'Coroa', 'Cara' ],  
[ 'Coroa', 'Coroa', 'Coroa' ],  
[ 'Coroa', 'Coroa', 'Coroa' ],  
[ 'Cara', 'Cara', 'Cara' ],  
[ 'Coroa', 'Coroa', 'Coroa' ],  
[ 'Coroa', 'Coroa', 'Coroa' ],  
[ 'Cara', 'Cara', 'Cara' ]
```

]

Info: End process (2:20:59 AM)

TADS – JavaScript

função:

exercício:

como desenvolver o simulador:

- construir uma function entrada de dados via console –

 - numero de moedas e numero de tentativas (p/simulador)

- construir uma function geraMoedas que deve aleatoriamente sortear Cara (F), Coroa (K) –
para o lançamento de cada moeda

- construir uma function geraJogada deve chamar geraMoedas gerar a sequencia de jogadas

- construir uma function comparaJogadas deve comparar a Jogada inicial com as demais (sim)

- construir uma function simula_jogoMoedas para simular o jogo

 - chamar a function geraJogadas

 - chamar a function comparaJogadas

 - se ocorrer acerto publicar senão passar para a próxima

função:

exercício:

como desenvolver o simulador:

- construir uma function simulaJogoMoedas para simular o jogo

 - chamar a function geraJogadas

 - chamar a function comparaJogadas

 - se ocorrer acerto publicar senão passar para a próxima

- construir uma function publicaResultado

 - deve ser chamada pela simulaJogoMoedas

 - publica as jogadas

 - publica acerto ou não

fim

TADS – JavaScript

função:

exercício:

restrições –

evitar que o simulador repita palpites já realizados, ou seja, para uma sequência, ou jogada de moedas repetida, a jogada deve ser descartada.

quem vai entrar com a jogada inicial agora, é o usuário ou jogador, via tela de comando.

o jogador deve receber a sequência de possíveis jogadas e entrar com a jogada escolhida e a quantidade de tentativas realizadas pelo simulador.

a saída deve ser a mesma, ou seja, quando o simulador acertar, dentro do limite imposto pelo jogador, ele informa quantas e quais jogadas foram realizadas. Ou se não acertou dentro do limite proposto pelo jogador.

TADS – JavaScript

função:

exercício:

restrições –

evitar que o simulador repita palpites já realizados, ou seja, uma sequência de moedas repetida, a jogada deve ser descartada.

via tela de comando, o jogador deve receber a sequência de possíveis jogadas e entrar com a jogada escolhida.

construir uma página em HTML simples que permita a entrada e saída de dados.

pense numa página que seja amigável com o jogador!

TADS - JavaScript

Exercício :

2 –

```
JS moedasaleatorias.js > ...
1  function jogamoedas(){
2      const resultados = [];
3      let numerodemoedas = 3;
4      for (let i = 0; i < numerodemoedas; i++){
5          resultados.push(Math.random() < 0.5 ? 'Cara' : 'Coroa');
6      }
7      return resultados;
8  }
9  function simularjogo(numerojogadas) {
10     const historicojogada = [];
11     for (let i = 0; i < numerojogadas; i++) {
12         const resultadojogada = jogamoedas(3);
13         historicojogada.push(resultadojogada);
14         console.log(`jogada ${i+1}: ${resultadojogada.join(',')}`);
15     }
16     return historicojogada;
17 }
18 const historico = simularjogo(10);
19 console.log("historico completo: ", historico);
20
```

função:

```
10
11  function geraMoedas() {
12      const vetMoedas = [];
13      for(let i=0; i< nuMoedas; i++){
14          vetMoedas.push(Math.random() < 0.5 ? 'Cara' : 'Coroa');
15      }
16      return vetMoedas;
17  }
18  function geraJogadas(){
```


função

exerc

const

JS JM_Funcs3.js

JS JM_Funcs3-2.js X

JS JM_Funcs3-2.js >  comparaJogadas

```
1  let prompt = require("prompt-sync")();
2  let nuMoedas;
3  let nuTenta;
4  let i=0;
5  let l = 0;
6  function entDadosUsuario() {
7      nuMoedas = Number(prompt("entre com o número de moedas a ser lançadas: "));
8      nuTenta = Number(prompt("entre com o numero de tentativas para que o simulador acertar a sequência gerada: "));
9  }
10
11 function geraMoedas() {
12     const vetMoedas = [];
13     for(let i=0; i< nuMoedas; i++){
14         vetMoedas.push(Math.random() < 0.5 ? 'Cara' : 'Coroa');
15     }
16     return vetMoedas;
17 }
18 function geraJogadas(){
19     let j = 0;
20
21     const historicoJogada = [];
22     while (j < nuTenta) {
23         const resultadoJogada = geraMoedas()
24         historicoJogada.push(resultadoJogada);
25         l = 0
26         comparaJogadas(resultadoJogada);
27         console.log("l: ", l );
28         console.log(`jogada ${j}: ${resultadoJogada.join(',')}`);
29
30         if(l == nuMoedas) {
31             j = nuTenta;
32         }
33     }
```

TADS – JavaScript

função:

exercício:

construir um simulador semelhante, trocando moedas por dados e jogar 2 ou 3 dados.

TADS – JavaScript

bibliografia

SILVA, M. S., JavaScript Essencial, Novatec, 2022

MORRISON, M., Use a Cabeça JavaScript, ed O'Reilly, Alta Books, 2009

ACKERMAN, P., JavaScript the comprehensive guide, Ed. Rheinwerk, 2022

W3C- resource: <https://www.w3resource.com/javascript/javascript.php> - 20/05/2024

CISCOACADEMY - <https://www.netacad.com/>

(<https://www.netacad.com/launch?id=062c7386-e936-452f-a584-65e0a3355ced&tab=curriculum&view=9ed3b581-161c-5b13-84c8-2074e7584032>)

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript> _ Site da Mozilla Developers

Alguns exercícios de livros de C