# Teste de Desenvolvedor JavaScript Júnior

## Instruções

1. Leia cada questão com atenção e implemente as soluções em JavaScript.
2. Certifique-se de comentar seu código explicando o que cada parte faz.
3. Envie a implementação para análise em um repositório no GitHub.
4. É desejável a implementação de testes unitários.

## Questão 1: Fila (Queue)

Implemente uma classe `Queue` utilizando arrays. Sua classe deve ter os seguintes métodos:

- `enqueue(item)`: adiciona um item ao final da fila.
- `dequeue()`: remove o item no início da fila e retorna seu valor.
- `peek()`: retorna o valor do primeiro item da fila sem removê-lo.
- `isEmpty()`: retorna `true` se a fila estiver vazia e `false` caso contrário.

**Exemplo de uso esperado:**
```javascript
const queue = new Queue();
queue.enqueue(10);
queue.enqueue(20);
console.log(queue.peek()); // 10
console.log(queue.dequeue()); // 10
console.log(queue.isEmpty()); // false
```

## Questão 2: Anagramas

Escreva uma função `areAnagrams(str1, str2)` que receba duas strings e retorne `true` se elas forem anagramas (isto é, contiverem os mesmos caracteres na mesma quantidade) ou `false` caso contrário.

**Exemplo:**
```javascript
areAnagrams("listen", "silent"); // true
areAnagrams("hello", "world"); // false
```

## Questão 3: Busca Binária

Implemente a função `binarySearch(arr, target)` que utilize o algoritmo de busca binária para encontrar um número em um array ordenado. A função deve retornar o índice do

número caso ele seja encontrado, ou `-1` caso contrário.

**Exemplo:**
```javascript
const arr = [1, 3, 5, 7, 9, 11];
console.log(binarySearch(arr, 5)); // 2
console.log(binarySearch(arr, 10)); // -1
```

### Questão 4: Número Repetido
Escreva uma função `findFirstDuplicate(arr)` que receba um array de números inteiros e retorne o primeiro número duplicado encontrado. Caso não existam duplicados, retorne `-1`.

**Exemplo:**
```javascript
findFirstDuplicate([2, 5, 1, 2, 3, 5]); // 2
findFirstDuplicate([1, 2, 3, 4]); // -1
```

### Questão 5: Inversão de String
Implemente uma função `reverseString(str)` que inverta uma string.

**Exemplo:**
```javascript
reverseString("hello"); // "olleh"
reverseString("javascript"); // "tpircsavaj"
```

### Questão 6: Soma de Pares
Escreva uma função `hasPairWithSum(arr, sum)` que receba um array e um número `sum`. A função deve retornar `true` se existir um par de números no array cuja soma seja igual ao número dado, ou `false` caso contrário.

**Exemplo:**
```javascript
hasPairWithSum([1, 2, 3, 9], 8); // false
hasPairWithSum([1, 2, 4, 4], 8); // true
```

### Questão 7: Palíndromo
Implemente uma função `isPalindrome(str)` que verifica se uma string é um palíndromo (lê-se da mesma forma de trás para frente).

**Exemplo:**
```javascript
isPalindrome("radar"); // true
isPalindrome("javascript"); // false
```

## Questão 8: Criando um Endpoint com Express

Crie um pequeno servidor utilizando o framework **Express**. O servidor deve ter um endpoint `POST /anagrams` que recebe no corpo da requisição duas strings (`str1` e `str2`) e verifica se elas são anagramas. O servidor deve retornar um JSON com o seguinte formato:

- Se as strings forem anagramas:
  ```json
  { "areAnagrams": true }
  ```
- Se as strings não forem anagramas:
  ```json
  { "areAnagrams": false }
  ```

**Requisitos:**
1. O servidor deve rodar na porta `3000`.
2. O corpo da requisição deve ser enviado no formato JSON.
3. Utilize o método `areAnagrams` que você criou na **Questão 2** para validar as strings.

**Exemplo de requisição e resposta:**

**Requisição:**
```bash
POST http://localhost:3000/anagrams
Content-Type: application/json

{
  "str1": "listen",
  "str2": "silent"
}
```

**Resposta:**
```json
{ "areAnagrams": true }
```

**Dica:**
Caso não tenha o Express instalado, você pode rodar o comando abaixo para instalá-lo:
```bash
npm install express
```

## Critérios de Avaliação

1. Organização e clareza do código.
2. Uso correto das estruturas de dados.
3. Correção dos algoritmos.
4. Uso eficiente de recursos computacionais.
5. Tratamento de erros no endpoint (Questão 8).