

Desafio: Fatorial Desajeitado

1 / 1 - Fatorial desajeitado

// IMPORTANTE: As funções "gets" e "print" são acessíveis globalmente, onde:

// - "gets" : lê UMA linha com dado(s) de entrada (inputs) do usuário;

// - "print": imprime um texto de saída (output) e pula uma linha ("\n") automaticamente.

```
let lines = gets().split("\n");
```

```
let n = parseInt(lines.shift());
```

```
if (n <= 2) {
```

```
    print(n);
```

```
}
```

```
let nZero = n;
```

```
let soma = 0;
```

```
const varia = -1;
```

```
while (n >= 4 ) {
```

```
    if (n == nZero) {
```

```
        soma += parseInt(n*((n-1) / (n-2)));
```

```
        soma += (n-3);
```

```
        n = n - 4;
```

```
    } else {
```

```

    soma += varia * parseInt(n*((n-1) / (n-2)))
    soma += (n-3);
    n = n - 4;
  }}

```

```

if (n == 3) {
    soma += varia * parseInt(n*((n-1) / (n-2)));
} else if (n == 2) {
    soma += varia * (n * (n-1));
} else if (n == 1) {
    soma += varia * n
}

```

```

print(soma);

```

// TODO: Calcule o valor do fatorial de "n", atribuindo o retorno a "r".

Desafio: Checagem de Palindromos

1 / 1 - Checagem de Palíndromos

// IMPORTANTE: As funções "gets" e "print" são acessíveis globalmente, onde:

// - "gets" : lê UMA linha com dado(s) de entrada (inputs) do usuário;

// - "print": imprime um texto de saída (output) e pula uma linha ("\n") automaticamente.

```

let resultado = gets();

```

```

print(checaPalindromo(resultado));

```

```

function checaPalindromo(entrada){

```

```

    const resultado = entrada.split('').reverse().join('');

```

```

    return (resultado === entrada) ? 'TRUE' : 'FALSE';

```

```
}
```

```
//TODO:Retorne TRUE ou FALSE, caso a "palavra" seja ou não um Palíndromo.
```

Desafio: Quadrados Perfeitos

1 / 1 - Quadrados Perfeitos

```
// IMPORTANTE: As funções "gets" e "print" são acessíveis globalmente, onde:
```

```
// - "gets" : lê UMA linha com dado(s) de entrada (inputs) do usuário;
```

```
// - "print": imprime um texto de saída (output) e pula uma linha ("\n") automaticamente.
```

```
let lines = gets().split("\n");
```

```
let n = parseInt(lines.shift());
```

```
const numSquares = n => {
```

```
  if (n < 4) return n;
```

```
  let best = Infinity;
```

```
  for (let x = 2; x <= Math.floor(Math.sqrt(n)); ++x) {
```

```
    let k = Math.floor(n / (x * x));
```

```
    best = Math.min(best, k + numSquares(n - k * x * x));
```

```
  }
```

```
  return best;
```

```
};
```

```
console.log(numSquares(n))
```

```
//TODO: Com base no valor total, retorne o menor número de quadrados perfeitos.
```

Desafio: Matriz Par e Impar

1 / 1 - Matriz Par e Impar

```
// IMPORTANTE: As funções "gets" e "print" são acessíveis globalmente, onde:  
// - "gets" : lê UMA linha com dado(s) de entrada (inputs) do usuário;  
// - "print": imprime um texto de saída (output) e pula uma linha ("\n") automaticamente.  
// - ".split()": transforma a entrada em um array separado por aspas simples
```

```
let arr = gets().split("");
```

```
let arrVazio = arr;
```

```
function orderArray(arr) {  
  const newArr = []  
  arr.forEach((item) => item%2===0? newArr.unshift(item) : newArr.push(item))  
  return newArr  
}
```

```
print(orderArray(arr))
```

```
// TODO: crie a condição necessária para que o array fique ordenado como o exercício pede
```

Desafio: Validação de Parênteses

1 / 1 - Validação de Parênteses

```
// IMPORTANTE: As funções "gets" e "print" são acessíveis globalmente, onde:  
// - "gets" : lê UMA linha com dado(s) de entrada (inputs) do usuário;  
// - "print": imprime um texto de saída (output) e pula uma linha ("\n") automaticamente.
```

```
var isValid = function(s) {  
  const stack = {  
    c: ["(", ")"],  
    p: ["(", ")"],  
    k: ["{", "}"]  
  };  
  for( let key in stack ) {  
    if ( s[0] === stack[key][0] ) {  
      if ( s[0] === stack[key][0] && s[1] === stack[key][1] ) {  
        return true;  
      }  
    }  
    else {  
      return false;  
    }  
    break;  
  }  
}  
print( isValid(gets()) );
```