



Curso de Computação  
Disciplina: **Tecnologias de Desenvolvimento de Interfaces Gráficas**  
Turma: **001**  
Professor: Demetrio Gomes Mestre

## Atividade Avaliativa III

**01)** Anderson é um aluno muito esforçado e estudioso. Ele se envolve em várias atividades da universidade. O final do ano está chegando, e com ele, o mês (dezembro) da apresentação de seu TCC. Ele, cheio de atividades, precisa saber se será possível apresentar seu tcc antes do Natal! Mas, antes da realização de sua apresentação, seu trabalho precisa ser revisado por seu orientador russo, o Prof. Florovsky.

### Entrada

A entrada é composta por um valor E ( $0 < E < 25$ ) que representa o dia no qual foi entregue o tcc para revisão. Um valor D ( $0 < D < 25$ ) representando a data final que o documento de TCC deve ser entregue para revisão.

### Saída

Mostre, para cada caso de teste, se Anderson apresentará ou não o tcc. A única possibilidade da entrega não ser realizada na data é por falta de orientação de Florovsky. Caso não seja possível, imprima "Eu odeio o prof. Florovsky!". Caso o trabalho seja entregue em até 3 dias antes do prazo final, imprima "Muito bem! O aluno está apto a apresentar até o natal!". Caso contrário, sendo muito próximo da data limite imprima "O trabalho está muito ruim!". Nesse último caso, serão adicionados mais dois dias para a realização das correções e, se a data final for menor que a data 24 (véspera do natal), ele poderá apresentar, sendo impresso "TCC Apresentado!", caso contrário imprima "Não deu! Só no próximo ano agora."

Exemplos de Entrada	Exemplos de Saída
13 19	Muito bem! O aluno está apto a apresentar até o natal!
22 23	O trabalho está muito ruim! Não deu! Só no próximo ano agora.
21 22	O trabalho está muito ruim! TCC Apresentado!

**02)** Imprima os seguintes campos usando o operador de desestruturação: autor (suite) - CEP - cidade (lat, lng). Imprima usando o recurso Template String.

Saída esperada: Demetrio JS Mestre: 581485-148 - Campina Grande (13.51, 52.54)

**Código:**

```
const usuario: any = {
  "autor": "Demetrio Mestre",
  "endereço": {
    "rua": "Rua Qualquer",
    "suite": "JS",
    "cidade": "Campina Grande",
    "CEP": "581485-148",
    "geo": {
      "lat": "13.51",
      "lng": "52.54"
    }
  }
}

const resultado = `Coloque seu código aqui`;
console.log(resultado);
```

**03)** Escreva uma função arrow que recebe um valor de X e imprima a soma dos 5 pares consecutivos a partir de X, inclusive o X, se for par. Se o valor de entrada for 4, por exemplo, a saída deve ser 40, que é o resultado da operação: 4+6+8+10+12, enquanto que se o valor de entrada for 11, por exemplo, a saída deve ser 80, que é a soma de 12+14+16+18+20. Atenção: Você deve usar uma função de callback para realizar o somatório.

Entrada

O valor de X.

Saída

Imprima a saída conforme a explicação acima e o exemplo abaixo.

Entrada: 4 => Saída: 40

Entrada: 11 => Saída: 80

**04)** Escreva duas funções arrow usando a técnica de **promises (Javascript ES6)**. A primeira função, **colocarTodasLetrasEmMaiusculoEm500ms**, receberá um único parâmetro e o colocará em maiúsculo depois de 500 milissegundos. Verifique, dentro da função, se o parâmetro se trata de um tipo String. Se o parâmetro não for do tipo String, a promise deve rejeitar o parâmetro. A segunda função, **inverteTodasLetras()**, realizará uma inversão das letras da String. Atenção: você deverá utilizar encadeamento de funções.

**05)** Transforme as funções desenvolvidas na questão 04 para usarem a técnica **async/await** ao invés de só usar **promises**.

Bom trabalho!