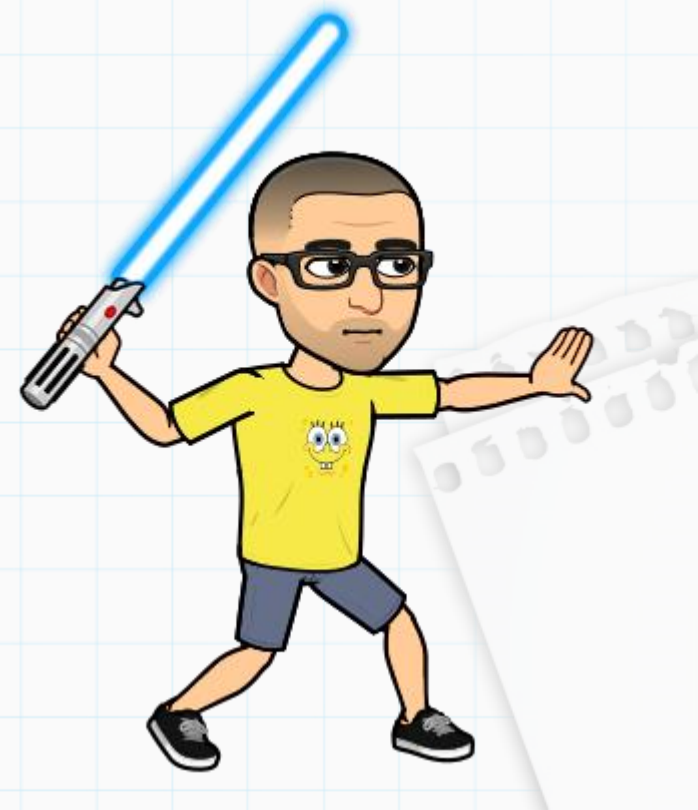


Programação De Computadores

Professor : Yuri Frota

www.ic.uff.br/~yuri/prog.html

yuri@ic.uff.br



Subrotinas e Funções

Exercício 1): Faça um programa que informe o status de 5 alunos a partir da sua média de acordo com a tabela a seguir:

Nota acima de 6 → “Aprovado”

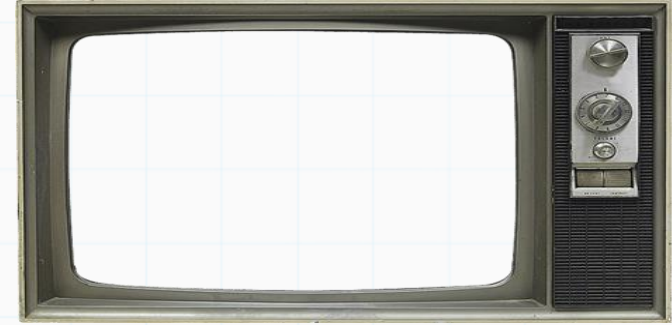
Nota entre 4 e 6 → “Verificação Suplementar”

Nota abaixo de 4 → “Reprovado”

use uma função que receba o nome e a média e imprimir o status equivalente

Ex: execução:

```
nome:Deku
média:5.4
Deku : VS :|
nome:Ka-chan
média:2.2
Ka-chan : Reprovado :(
nome:Uraraka
média:9.7
Uraraka : Aprovado :)
nome:All Might
média:6.6
All Might : Aprovado :)
nome:Tenya
média:10
Tenya : Aprovado :)
```



FUNÇÃO(...)

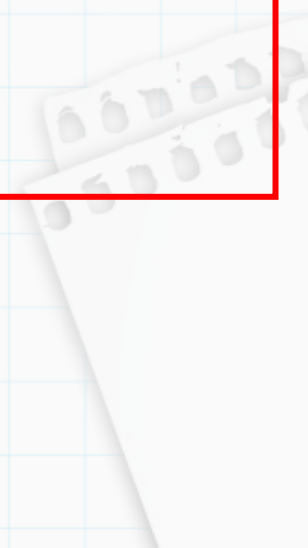
imprime status

Programa Principal

I percorre alunos

Le nome e média do I-ésimo aluno

FUNÇÃO(...)



Subrotinas e Funções

Exercício 1): Faça um programa que informe o status de 5 alunos a partir da sua média de acordo com a tabela a seguir:

Nota acima de 6 → “Aprovado”

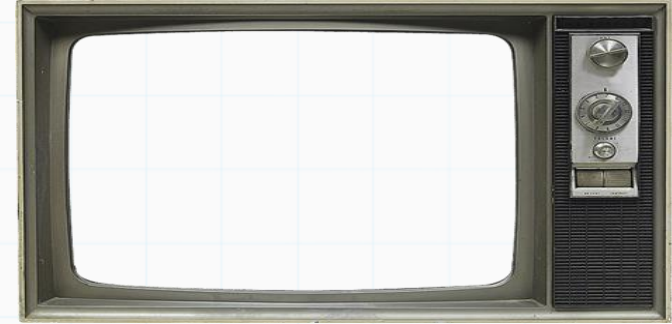
Nota entre 4 e 6 → “Verificação Suplementar”

Nota abaixo de 4 → “Reprovado”

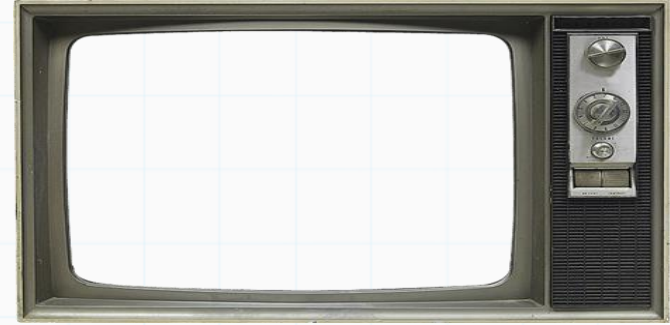
use uma função para determinar e imprimir o status

```
1 def status(nome,media):
2     if (media>6):
3         print(nome,":","Aprovado :)")
4     elif (media>4):
5         print(nome,":","VS :|")
6     else:
7         print(nome,":","Reprovado :(")
8
9     for i in range(5):
10        nome  = input("nome:")
11        media = float(input("média:"))
12        status(nome, media)
```

[código](#)



Subrotinas e Funções



Exercício 2): Receba do usuário uma matriz de inteiros M $n \times n$. Depois crie uma função que imprima o número de linhas nulas da matriz M . M deve ser criada no programa principal e uma função será usada para determinar e imprimir o número de linhas nulas de M .

Ex. execução:

$n=3$

```
[0, 0]:0  
[0, 1]:1  
[0, 2]:0  
[1, 0]:0  
[1, 1]:0  
[1, 2]:0  
[2, 0]:3  
[2, 1]:0  
[2, 2]:0
```



matriz

```
[0, 1, 0]  
[0, 0, 0]  
[3, 0, 0]
```

1 linhas nulas

FUNÇÃO(...)

I percorre linhas de M

J percorre colunas de M

Determina se a I-ésima linha de M é nula

Se I-ésima linha de M nula então incrementa contador

Imprime contador

Programa Principal

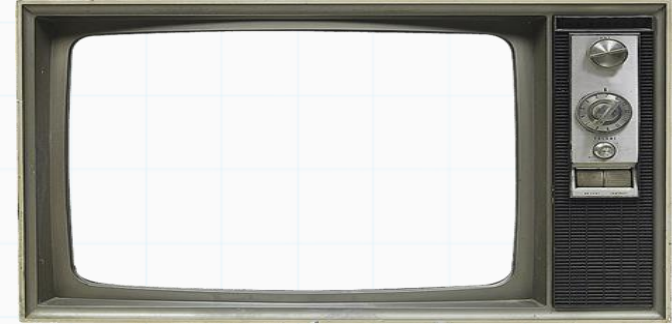
Recebe matriz M

FUNÇÃO(...)

Subrotinas e Funções

Exercício 2): Receba do usuário uma matriz de inteiros M $n \times n$. Depois crie uma função que imprima o número de linhas nulas da matriz M . M deve ser criada no programa principal e uma função será usada para determinar o número de linhas nulas de M .

```
1 def linhas_nulas(M,n):
2     num = 0
3     for i in range(n):
4         nula = True
5         for j in range(n):
6             if (M[i][j] != 0):
7                 nula = False
8                 break
9         if (nula):
10             num = num+1
11     print(num, ' linhas nulas')
13 n= int(input("n="))
14 matriz = []
15 for i in range(n):
16     linha = []
17     for j in range(n):
18         linha.append(int(input('['+str(i)+', '+str(j)+']:')))
19     matriz.append(linha)
20 for i in range(n):
21     print(matriz[i])
22
23 linhas_nulas(matriz,n)
```



[código](#)

Subrotinas e Funções

Exercício 2): Receba um número inteiro positivo n do usuário e calcule seu fatorial SEM USAR LAÇOS.



Sabendo que :

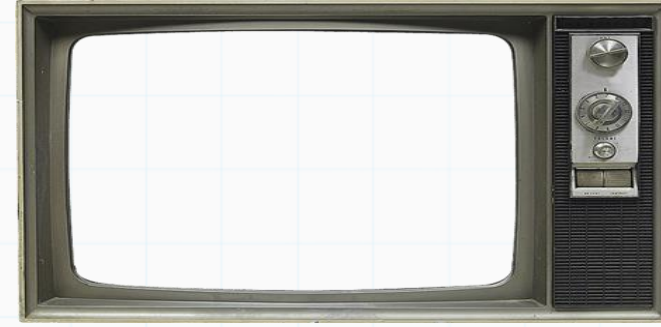
$$n! = \begin{cases} 1 & \text{for } n = 0 \text{ or } n = 1 \\ n \times (n - 1)! & \text{for } n \geq 2 \end{cases}$$

Caso base (parada da recursão)

recursão



RECURSÃO

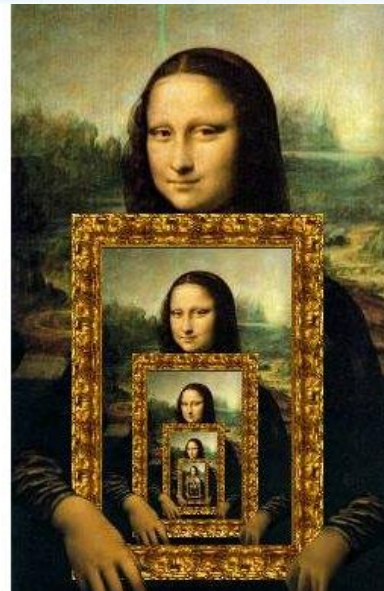


FUNÇÃO(n)
...
Retorne FUNÇÃO(?)

Programa Principal

Recebe n

FUNÇÃO(n)



Subrotinas e Funções

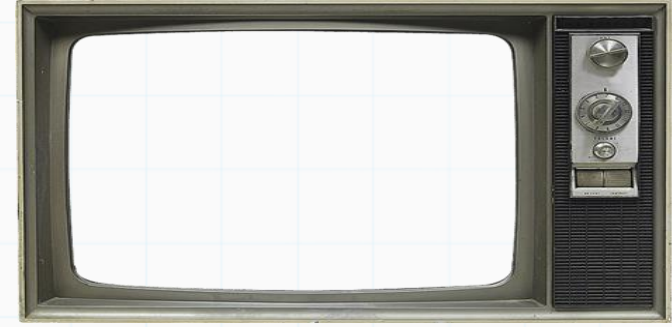


Exercício 2): Receba um número inteiro positivo n do usuário e calcule seu fatorial usando uma função recursiva.

Sabendo que :

```
1 def fatorial(n):  
2     if (n <= 1):  
3         return 1  
4     else:  
5         return n * fatorial(n - 1)  
6  
7 n = int(input("n:"))  
8 print(fatorial(n))
```

Até a próxima



Slides baseados no curso de Vanessa Braganholo

