



Study Jam

PROG1 - Minitest 1

 github.com/gdscuporto



```
filterByOrg = filterByOrg ? study.lead_organization === filterByOrg : true  
filterByStatus = filterByStatus ? study.status === filterByStatus : true  
function filterStudies({ studies, filterByOrg, filterByStatus }) {  
  return studies.filter(study => filterByOrg && filterByStatus)}
```

Exemplo 1

Produção de refrigerantes

Escreva um programa que permite obter estatísticas de uma linha de enchimento de refrigerantes. Para cada unidade, é indicado 1 valor (float), correspondente ao tempo de enchimento da lata, em segundos.

Com base nestes valores lidos, o programa deve calcular as seguintes estatísticas: os tempos máximo e mínimo (com 2 casas decimais) de enchimento, bem como o tempo total de enchimento das unidades (em minutos, segundos e milésimos de segundo).

Considera apenas as unidades que demoraram mais de 4s e menos de 6s a ser enchidas.

Teste o seu programa com o ficheiro refrigerantes.txt

Exemplo 1

Produção de refrigerantes

Resultado esperado:

```
Tempo minimo de enchimento: 4.01s  
Tempo máximo de enchimento: 5.94s  
Tempo total: 1 minuto(s), 10 segundo(s), 761 milissegundos.
```


Exemplo 2

Velocidade do vento

Implemente um programa que permita analisar a amplitude da velocidade do vento numa dada região de Portugal. Para cada mês, devem ser lidos 3 valores: ID do mês (inteiro), velocidade máxima (float) e velocidade mínima (float).

O programa deverá calcular a amplitude da velocidade mensal (calculada como a diferença entre a velocidade máxima e a velocidade mínima num mês) e indicar a maior amplitude e em que ano e mês ocorreu.

O ID do mês é inicializado em 1, correspondente a janeiro de 2018, e é incrementado sequencialmente; por exemplo, mês 13 significa janeiro de 2019.

Teste o seu programa com o ficheiro vento.txt

Exemplo 2

Velocidade do vento

Resultado esperado:

```
Maior amplitude de velocidade: 1.48  
Registada no mes 7 do ano 2020
```

Exemplo 3

Operações vetoriais R^3

Pretende-se implementar um programa para o cálculo de operações com vetores em R^3 . Para isso, o programa deve ler uma letra correspondente ao tipo de operação e caso essa letra seja:

- A - lê 6 números reais ($u_x, u_y, u_z, v_x, v_y, v_z$) e calcula a adição dos vetores u e v
- B - lê 4 números reais (c, u_x, u_y, u_z) e calcula o produto de um escalar pelo vetor u
- C - lê 6 números reais ($u_x, u_y, u_z, v_x, v_y, v_z$) e calcula o produto escalar dos vetores u e v

O programa imprime o resultado de cada operação e o total de operações de cada tipo. Devem ser contabilizadas e apresentadas o número de vezes que cada operação foi realizada.

Teste o seu programa com o ficheiro vetores.txt

Exemplo 3

Operações vetoriais \mathbb{R}^3

Resultado esperado:

```
u . v = -30.024403
u . v = -4.309000
u + v = (0.410000, 7.550000, 0.230000)
...
c * u = (30.096001, -32.771202, -11.756800)
c * u = (-13.902301, 15.492600, 42.065998)
c * u = (2.643200, -12.723200, 23.340799)
```

Adicao de vetores: 18

Produto de um escalar: 26

Produto escalar de vetores: 20

Exemplo 4

Combustível para a viagem

Crie um programa que calcule a quantidade de combustível mínimo necessária para efetuar uma viagem de transporte de mercadorias, sabendo que essa quantidade é influenciada pelo tipo de embarcação, o peso da mercadoria e a duração da viagem. Para isso, o programa deve ler uma letra correspondente ao tipo de embarcação e, caso esse caracter seja:

- A - lê 2 números reais (duração e peso) e calcula o combustível da embarcação A:
 $CombustivelA = CB + duracao * peso/10$
- B - lê 2 números reais (duração e peso) e calcula o combustível da embarcação B:
 $CombustivelB = CB + duracao * (peso/1500)^2$
- C - lê 1 número real (duração) e calcula o combustível da embarcação C:
 $CombustivelC = CB + duracao^3$

1/2

Exemplo 4

Combustível para a viagem

A soma de todo o combustível também deve ser calculada e mostrada.

Nota: define CombustivelBase (CB) como sendo 350.

2/2

Exemplo 4

Combustível para a viagem

Resultado esperado:

```
351.00  
409.20  
...  
1250.07  
  
30006.36 (total)
```