

MC102QR - Algoritmos e Programação de Computadores

Lab03: Homem Aranha De Volta ao Lab

Prazo: 17 de Abril de 2022.

Peso na nota: 1 (2,44%)

Para criar o antídoto para o vilão Duende Verde e acabar com os desastres que ele causa em Nova Iorque, Peter Parker e seu amigo Ned precisaram fazer várias baterias de testes de uma solução para saber a quantidade certa do *Vibranium* (elemento fornecido por Happy Hogan em quantidade limitada) que deve ser utilizada. Para facilitar e agilizar esse processo, Peter pensou em fazer um código que calculasse a quantidade do *Vibranium* para cada um dos frascos que ele pretende testar a solução, assim como a quantidade total de *Vibranium* para encomendar com Happy, somando a quantidade em todos os frascos. Estes frascos são marcados numericamente por um índice i de forma sequencial (1,2,3,...).



Para fazer este cálculo, Peter vai usar dois valores auxiliares que, juntamente com a quantidade de frascos q , serão parâmetros de entrada. São eles: a Taxa de *Vibranium* T , que é a quantidade do *Vibranium* para determinada bateria de testes; e a quantidade C do solvente (valor inteiro) utilizado por Peter para poder manusear o *Vibranium*, que é um valor igual em todas os frascos de uma bateria de testes.

Assim, o cálculo precisará ser feito, para cada frasco, da seguinte forma: $T * i + T * C$. Desta forma, o primeiro trecho ($T * i$) vai determinar a quantidade de acordo com o frasco e o segundo trecho ($T * C$) indica a perda de *Vibranium*.

A partir disso, o método que Peter pensou para desenvolver esse código foi calcular a quantidade de *Vibranium* ***Vi*** utilizada em cada frasco ***i***, mostrando a informação na tela, e também mostrar a quantidade parcial ***Vp*** de *Vibranium* já utilizada a cada vez que um novo frasco fosse calculado. Ao final, o código mostra o total de *Vibranium* ***Vt*** nessa bateria de testes. Por exemplo uma bateria de testes com 2 frascos, os resultados manuais seriam:

- Frasco 1 resultou em 95.58ml de *Vibranium*, soma parcial de 98.50ml
- Frasco 2 resultou em 96.12ml de *Vibranium*, soma parcial de 191.70ml (isto é, a quantidade deste frasco + a do frasco anterior)
- Total de *Vibranium* resultou em 195.90ml

Ned, que também sofreu ataques do Duende, ficou responsável por algumas partes desse processo, porém ele precisa dos valores que Peter encontrou em seus testes. Ned precisa calcular uma progressão aritmética ***PAn*** de acordo com uma constante inteira ***n*** (resultante de um cálculo que Ned fez). A progressão vai somar ***n + n + n + ...*** até que a soma atinja (ou até a última vez antes de atingir) o valor total do *Vibranium* ***Vt*** encontrado por Peter. Além disso, para anotações de Ned, é necessário um contador ***m*** (com valor inicial sendo 0) dizendo quantas vezes foi necessário fazer essa soma, isto é, quantas iterações foram feitas. Usando o exemplo do valor encontrado por Peter no parágrafo anterior: Se ***n*** = 49, os resultados manuais seriam:

- soma = 49, contador = 1
- soma = 98, contador = 2
- soma = 147, contador = 3 (se aqui for somado 49 de novo, a soma ultrapassa o ***Vt***)
- valor final do contador = 3

No final para saber que os testes foram finalizados, eles precisam de uma confirmação do programa dizendo “BATERIA DE TESTES TERMINADA” para que eles possam preparar uma nova bateria de testes, caso precise.

Tarefa

Fazer um programa que cumpra as seguintes tarefas:

- Calcule a quantidade de *Vibranium* ***Vi*** para cada frasco de acordo com a seguinte fórmula:
 - **$V_i = T * i + T * C$**
- De acordo com a quantidade de *Vibranium* em cada frasco, calcule a quantidade parcial ***Vp*** e total ***Vt*** de *Vibranium* utilizada na bateria de testes
- Calcule a progressão aritmética ***PAn*** de acordo com a constante ***n*** e de acordo com os cálculos de *Vibranium*, fornecendo também o valor final do contador ***m***

Entrada

O código vai receber como entrada:

- Quantidade de frascos com as soluções ($1 \leq q \leq 100$);

- Taxa de *Vibranium* que vai ser usada nessa bateria de testes ($0.01 \leq T \leq 1.00$);
- Constante do solvente ($0 \leq C \leq 1000$);
- Constante da progressão aritmética ($1 \leq n \leq 100$).

Assim, o formato da entrada será:

[q]

[T]

[C]

[n]

Saída

- Quantidade de *Vibranium* (**Vi**) utilizada em cada frasco
- Somas parciais (**Vp**) de *Vibranium* (para cada índice de frasco)
- Soma total (**Vt**) de *Vibranium* usado em toda bateria de testes
- Valores da progressão aritmética de Ned (**PAn**)
- Valor final do contador **m**
- Mensagem final dizendo “BATERIA DE TESTES TERMINADA!”

A saída deverá respeitar o seguinte formato:

[f] [Vi] [Vp]

[f] [Vi] [Vp]

(...)

[Vt]

[PAn]

[PAn]

(...)

[m]

BATERIA DE TESTES TERMINADA

Obs. 1: O termo “(…)” nesse formato de saída representa a repetição da linha acima disso, então **esse termo “(…)” não deve ser impresso**, foi somente para ilustrar as iterações.

Obs. 2: Os valores *[Vi]*, *[Vp]* e *[Vt]* **devem ter duas casas decimais**.

Exemplos

Exemplo 1:

Entrada

```
5
0.56
100
49
```

Saída

```
1  56.56  56.56
2  57.12  113.68
3  57.68  171.36
4  58.24  229.60
5  58.80  288.40
288.40
49
98
147
196
245
5
BATERIA DE TESTES TERMINADA
```

Exemplo 2:

Entrada

```
2
0.98
134
90
```

Saída

```
1  132.30  132.30
2  133.28  265.58
265.58
90
180
2
BATERIA DE TESTES TERMINADA
```

Exemplo 3:

Entrada

```
8
0.32
251
100
```

Saída

```
1  80.64  80.64
```

```
2 80.96 161.60
3 81.28 242.88
4 81.60 324.48
5 81.92 406.40
6 82.24 488.64
7 82.56 571.20
8 82.88 654.08
654.08
100
200
300
400
500
600
6
BATERIA DE TESTES TERMINADA
```

Submissão da tarefa

Submeter no CodePost, na tarefa com o nome de *Lab 3 - Repetição simples*, o arquivo:

⇒ `lab3.py`: Arquivo onde deverá ser implementada a tarefa.

Após o prazo estabelecido para a atividade, será aberta uma tarefa *Lab 3 - Segunda Chance*, com prazo de entrega até o fim do semestre.

Está **proibido** o uso de função `sum()`, que já é integrada e pronta para uso no Python.

Peter e Ned vão voltar...