

MC102 - Algoritmos e Programação de Computadores

MC102 Horários Plano de desenvolvimento Cronograma
Oferecimentos anteriores



Ready...

Steady...

Slow!

Corrida de Caracóis

Com os campeonatos esportivos interrompidos, observar caracóis no quintal pode ser um passatempo... Podemos até classificar estes pequenos animais de acordo com sua velocidade e verificar se estariam aptos a participar de eventos internacionais como o [World Snail Racing Championship](#). Considerando o tempo em segundos que um caracol leva para realizar um percurso de 33 cm (13 polegadas), definimos:

- **Nível 0:** Se o tempo for menor do que 180 s
- **Nível 1:** Se o tempo for maior ou igual a 180 s e menor que 240 s
- **Nível 2:** Se o tempo for maior ou igual 240 s

Sua tarefa será ler as medidas de tempo de um grupo de caracóis e indicar (i) quantos animais estão em cada um dos níveis definidos, (ii) qual foi o tempo médio para a realização do percurso, (iii) qual foi a maior velocidade e (iv) qual foi a menor velocidade. Os valores de velocidade deverão ser expressos em cm/min.

Descrição da entrada

A entrada será composta por uma ou mais linhas contendo cada uma delas uma medida de tempo em segundos. O valor `-1` indicará o final da entrada. Veja o exemplo:

```
320
158
```

```
400
175
205
-1
```

Descrição da saída

A saída será composta por seis linhas que serão descritas a seguir. Note que os dados serão precedidos por strings explicativas, escritas propositalmente sem acentos. O tempo médio e os valores de velocidade devem ser formatados com uma casa decimal. Você pode conferir dicas sobre este tipo de formatação no enunciado da [Tarefa de Laboratório 02](#) e na página [Formatação em Python](#). Relembramos que as velocidades, em cm/min, devem ser calculadas considerando-se o percurso de 33 cm.

```
Caracois no nivel 0: 2
Caracois no nivel 1: 1
Caracois no nivel 2: 2
Tempo medio: 251.6 s
Velocidade maxima: 12.5 cm/min
Velocidade minima: 5.0 cm/min
```

Testes com o SuSy

No SuSy, para cada tarefa, criamos um conjunto de testes com arquivos de entrada `arq<i>.in` e para cada um deles temos uma saída esperada `arq<i>.res`. Para esta tarefa, os cinco primeiros testes abertos serão os listados na tabela abaixo. Consulte os arquivos no SuSy ou no pacote [aux06.zip](#) para ver o conteúdo dos demais testes abertos. Haverá também dois testes fechados.

Entrada		Saída	
arq01.in	258	Caracois no nivel 0: 0	arq01.res
	-1	Caracois no nivel 1: 0	
		Caracois no nivel 2: 1	
		Tempo medio: 258.0 s	
		Velocidade maxima: 7.7 cm/min	
		Velocidade minima: 7.7 cm/min	
arq02.in	180	Caracois no nivel 0: 0	arq02.res
	360	Caracois no nivel 1: 1	
	-1	Caracois no nivel 2: 1	
		Tempo medio: 270.0 s	
		Velocidade maxima: 11.0 cm/min	
		Velocidade minima: 5.5 cm/min	
arq03.in	150	Caracois no nivel 0: 1	arq03.res
	210	Caracois no nivel 1: 1	
	360	Caracois no nivel 2: 1	
	-1	Tempo medio: 240.0 s	
		Velocidade maxima: 13.2 cm/min	
		Velocidade minima: 5.5 cm/min	

arq04.in	205 317 195 160 -1	Caracois no nivel 0: 1 Caracois no nivel 1: 2 Caracois no nivel 2: 1 Tempo medio: 219.2 s Velocidade maxima: 12.4 cm/min Velocidade minima: 6.2 cm/min	arq04.res
arq05.in	307 198 176 340 145 500 203 -1	Caracois no nivel 0: 2 Caracois no nivel 1: 2 Caracois no nivel 2: 3 Tempo medio: 267.0 s Velocidade maxima: 13.7 cm/min Velocidade minima: 4.0 cm/min	arq05.res

Releia, se necessário as instruções para fazer os testes em [Testes com o SuSy](#).

Sobre o uso de listas

Você pode desenvolver uma solução eficiente para esta tarefa sem precisar armazenar as medidas de tempo em uma lista. O **uso de listas** e métodos que operam sobre estas listas é **permitido**, mas é importante que você saiba resolver o problema sem estes recursos adicionais.

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. O arquivo a ser submetido deve se chamar `lab06.py`. No link [Arquivos auxiliares](#) há um arquivo [aux06.zip](#) que contém todos os arquivos de testes abertos e seus respectivos resultados compactados.

Utilize o sistema SuSy com o mesmo login e senha que você utiliza para fazer acesso ao sistema da DAC. Se você não estiver inscrito corretamente, envie email para islene@ic.unicamp.br.

O limite máximo será de 30 submissões. Serão considerados os resultados da última submissão.

O peso desta tarefa é 2.

O prazo final para submissão é 17/05/2020.

A nota desta tarefa é proporcional ao número de testes que executaram corretamente, desde que o código esteja coerente com o enunciado. **A submissão de um código que não implementa o algoritmo requisitado, mas que exibe as saídas esperadas dos testes abertos a partir da comparação de trechos da entrada será considerada fraude e acarretará a atribuição de nota zero à média final da disciplina.**

Este exercício é uma variação de um dos problemas propostos na Olimpíada Cearense de Informática de 2014, cujo enunciado está disponível em [URI Online Judge | 1789](#). O [lab05](#) do primeiro semestre de 2018 solicitava a resolução deste problema em sua versão original.

A figura que ilustra esta tarefa foi obtida em <https://pixabay.com>.