

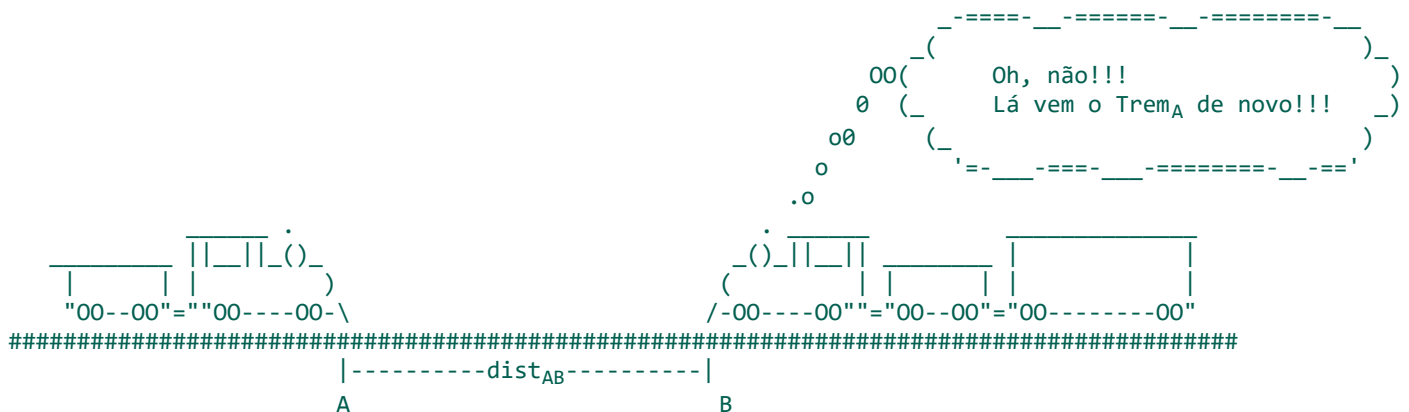
MC102 - Algoritmos e Programação de Computadores

MC102 Horários Plano de desenvolvimento Plano de aulas Oferecimentos anteriores

O encontro

Nesta tarefa, utilizaremos o tipo `float` para resolver um problema tradicional dos cursos introdutórios de Física. Duas estações de trem A e B estão separadas por uma distância $dist_{AB}$ medida ao longo da trajetória. Pela estação A passa um trem $Trem_A$ no sentido de A para B e simultaneamente passa por B um trem $Trem_B$ no sentido de B para A. Os trens $Trem_A$ e $Trem_B$ têm movimentos uniformes com velocidades v_A e v_B medidas em km/h e apresentando valores absolutos maiores do que zero. Você deverá determinar:

- o **instante** do encontro dos trens, medido em minutos, considerando como origem dos tempos o instante em que, simultaneamente, o $Trem_A$ passa pela estação A e o $Trem_B$ passa pela estação B.
- o **espaço**, medido em quilômetros, da parte dianteira dos trens no instante do encontro, adotando a estação A como origem dos espaços.



Trabalhando com o tipo float

Operações básicas Antes de escrever seu programa, vamos fazer alguns testes com a Python shell. Abra um terminal e o programa `python3`:

```
$ python3
Python 3.7.3
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Atribua números reais para algumas variáveis e escreva algumas operações. Veja os exemplos:

```
>>> velocidade_km_por_hora = 100.00
>>> velocidade_km_por_min = velocidade_km_por_hora / 60.00
>>> print(velocidade_km_por_min)
1.6666666666666667
```

Formatação Note que o valor foi exibido com 16 casas decimais. Podemos alterar o número de casas na saída utilizando o comando `format()`:

```
>>> format(velocidade_km_por_min, '.2f')
>>> '1.67'
>>> print("v =", format(velocidade_km_por_min, '.2f'), 'km/min')
v = 1.67 km/min
```

Abaixo temos duas outras maneiras de se obter esta saída. Para entender melhor o funcionamento destes comandos leia a página [Formatação em Python](#).

```
>>> "%.2f" % velocidade_km_por_min # estilo antigo - pode não funcionar nas próximas versões de Python
'1.67'
```

Limitações A exibição de um maior número de casas pode revelar resultados surpreendentes. Observe os valores 0.1 e 0.5 com 30 casas decimais:

```
>> format(0.1, '.30f')
'0.1000000000000000005551115123126'
>>> format(0.5, '.30f')
'0.500000000000000000000000000000'
```

Para entender a razão destes resultados, veja a seção [15. Floating Point Arithmetic: Issues and Limitations](#) da documentação de Python 3.7.

Descrição da entrada

A entrada do seu programa será composta por três linhas. A primeira conterá um `float` representando a distância `distAB` entre as duas estações, medida em quilômetros. A segunda e terceira linhas conterão, respectivamente, a velocidade em valor absoluto do `TremA` e a do `TremB`, em quilômetros por hora. Veja um exemplo:

75.00
50.00
25.00

Dica: você poderá ler estes valores com os comandos

```
dist_AB = float(input())
v_A = float(input())
v_B = float(input())
```

Descrição da saída

A saída deverá apresentar na sequência os seguintes valores em reais:

- **Instante do encontro:** instante em minutos, escrito com duas casas decimais.
- **Posição do encontro:** posição em quilômetros, escrita com duas casas decimais.

Para o exemplo acima, a saída será:

60.00 min
50.00 km

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 testes abertos serão os listados na tabela abaixo. Haverá também um teste fechado.

Entrada		Saída	
arq1.in	30.00 60.00 60.00	15.00 min 15.00 km	arq1.res
arq2.in	100.00 80.00 40.00	50.00 min 66.67 km	arq2.res
arq3.in	90.00 45.00 90.00	40.00 min 30.00 km	arq3.res
arq4.in	37.50 63.00 42.00	21.43 min 22.50 km	arq4.res

Leia instruções para fazer os testes em [Testes com o SuSy](#).

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. O arquivo a ser submetido deve se chamar `lab02.py`. No link [Arquivos auxiliares](#) há um arquivo [aux02.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 15 submissões. Serão considerados os resultados da última submissão.

O peso desta tarefa é 1.

O prazo final para submissão é 01/09/2019.

O diagrama que ilustra esta tarefa foi desenvolvido a partir do material disponível em www.asciart.eu.