


MC102 – Algoritmos e Programação de Computadores

			0
	7	8	9
	4	5	6
int	1	2	3
float	0	.	

Minissomadora

Nesta tarefa, vamos estudar comandos básicos de entrada e saída, comandos condicionais simples e manipulação de objetos do tipo `int` e `float`.

Tipo `int`

Objetos do tipo `int` armazenam valores inteiros. Observe os comandos abaixo, que podem ser testados na Python Shell.

```
$ python3
>>> i0 = -9
>>> type(i0)
<class 'int'>

>>> i1 = 3
>>> type(i1)
<class 'int'>

>>> print(i0 + i1)
-6

>>> type(i0 + i1)
```

```
<class 'int'>
>>>
```

Tipo float

Objetos tipo `float` armazenam valores reais. Veja estes outros exemplos:

```
$ python3
```

```
>>> f0 = -9.54
>>> type(f0)
<class 'float'>
```

```
>>> f1 = 3.0
>>> type(f1)
<class 'float'>
```

```
>>> type (f0 + f1)
<class 'float'>
```

```
>>> print(f0 + f1)
-6.539999999999999
```

O resultado `f0 + f1` pode parecer surpreendente. Se você quiser saber mais sobre armazenamento e aritmética de ponto flutuante, veja a seção [15. Floating Point Arithmetic: Issues and Limitations](#) da documentação sobre Python 3, mas você não precisa entender as limitações do armazenamento de números reais para realizar esta tarefa.

Podemos controlar a exibição do número de casas decimais utilizando a função de formatação `format`. No exemplo abaixo, a string `".2f"` indica que o resultado de `-9.54 + 3.0` será exibido com duas casas decimais.

```
$ python3
>>> print(format(-9.54 + 3.0, ".2f"))
-6.54
>>>
```

Combinando int e float

Podemos fazer operações combinando os dois tipos. Veja os exemplos:

```
$ python3
>>> type(3 + 4.0)
<class 'float'>
>>> type(3.0 + 4)
<class 'float'>
>>>
```

A tabela a seguir resume as combinações possíveis.

Operando 1	Operador	Operando 2	Resultado
int	+	int	int
int	+	float	float
float	+	int	float
float	+	float	float

Implementando a minissomadora

Você deverá implementar um programa em Python que soma dois números. Cada número virá precedido de uma letra `i` ou `f` indicando se este é um `int` ou `float`, como esquematizado a seguir:

```
<tipo_operando_1>
<valor_operando_1>
<tipo_operando_2>
<valor_operando_2>
```

Conforme o tipos dos operandos, o resultado será um `float` formatado com exatamente duas casas decimais ou um `int`.

Testes para 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 serão os seguintes:

Entrada		Resultado	
arq1.in	i 5 i 3	8	arq1.res
arq2.in	i 3 f 3.57	6.57	arq2.res
arq3.in	f 5.58 i 4	9.58	arq3.res
arq4.in	f 2.3589 f 6.3234	8.68	arq4.res

Dicas de Python 3 para esta tarefa

- Para ler o tipo de um operando e para armazenar seu valor, utilize comandos semelhantes a:

```
tipo = input()
obj_int = int(input())
obj_float = float(input())
```

- Para imprimir um objeto tipo float com duas casas decimais utilize:

```
print(format(obj_float, ".2f"))
```

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar **main.py**.

Todos os alunos matriculados até 8 de março estão inscritos. Se você entrou depois e/ou não estiver cadastrado corretamente, envie email para islene@ic.unicamp.br.

Observe o limite máximo de 10 submissões e que a última submissão deve ser correta.

O peso desta tarefa é 2.

O prazo final para submissão é 24/03/2018.
