

Nome do arquivo a ser entregue: **p07.py**

Obs.: Recomenda-se salvar o arquivo com certa frequência para não perder a digitação já feita caso haja uma falha na rede elétrica.

Preparação: faça o “download” dos arquivos **p07.py** e **fechadura.pyc** para dentro da “pasta pessoal”.

Você se lembra de “Tut”, a “tartaruga virtual” que usamos na Prática 1 para percorrer o labirinto e pegar o tomate? Vamos recordar alguns comandos usados para fazer Tut caminhar:

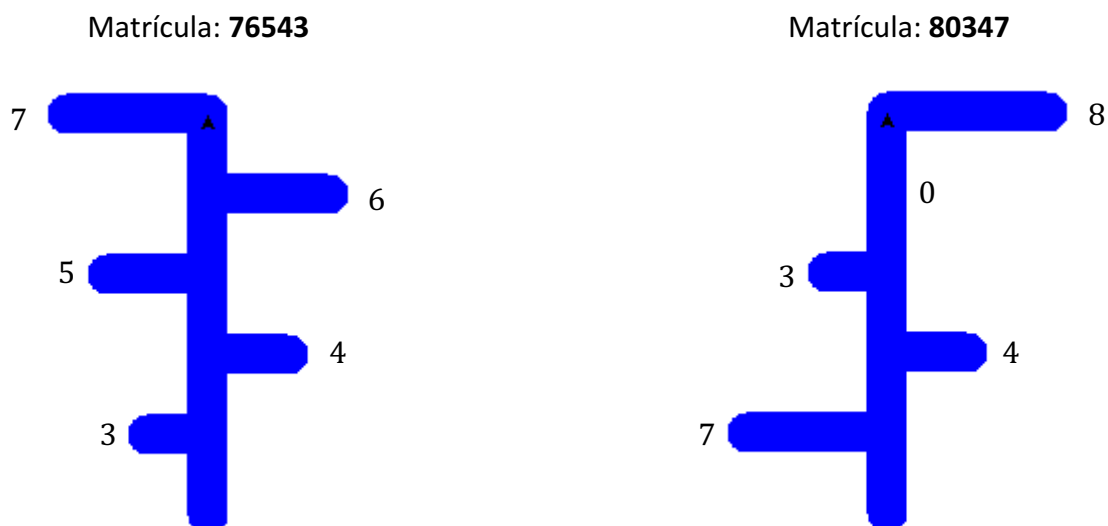
Comando	Efeito
<code>forward(n)</code>	andar n passos para frente
<code>backward(n)</code>	andar n passos para trás
<code>left(a)</code>	virar a graus para a esquerda
<code>right(a)</code>	virar a graus para a direita

Por exemplo, para fazer Tut andar 100 passos para frente, virar para a direita e andar mais 50 passos, você usaria a seguinte sequência de comandos:

```
Tut.forward( 100 )
Tut.right( 90 )
Tut.forward( 50 )
```

Nesta prática, você usará o rastro deixado pela tartaruga para desenhar uma “chave”. Essa chave deverá ser única para cada número de matrícula.

A chave terá “dentes” que dependerão dos dígitos presentes no número de matrícula. Dígitos pares resultam em dentes à direita da chave, e dígitos ímpares resultam em dentes à esquerda. Além disso, o comprimento de cada dente depende do valor do dígito. Dois exemplos são mostrados abaixo:



Para desenhar cada “dente” correspondente a um dígito d , a tartaruga deverá:

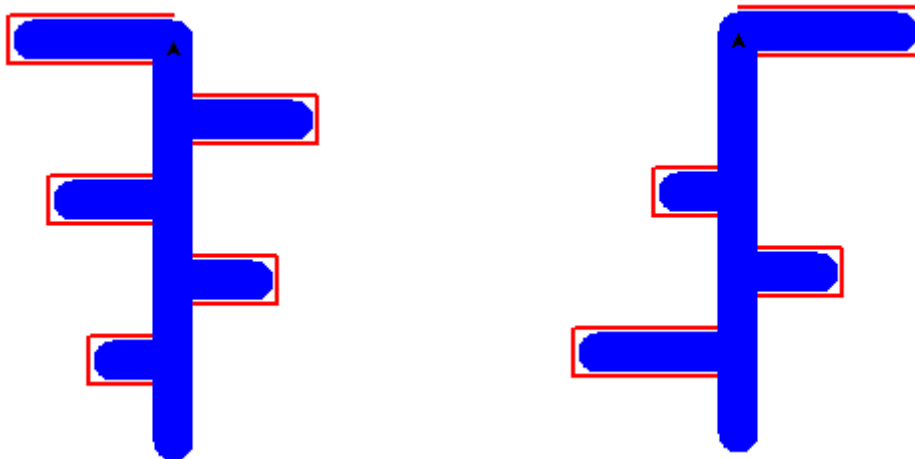
1. Caminhar 40 passos para frente;
2. Se d for par:
 - a. Virar 90° para a direita;
 - b. Andar $d * 10$ passos para frente;
 - c. Andar $d * 10$ passos para trás;
 - d. Virar 90° para a esquerda;

Se d for ímpar:

- a. Virar 90° para a esquerda;
- b. Andar $d * 10$ passos para frente;
- c. Andar $d * 10$ passos para trás;
- d. Virar 90° para a direita;

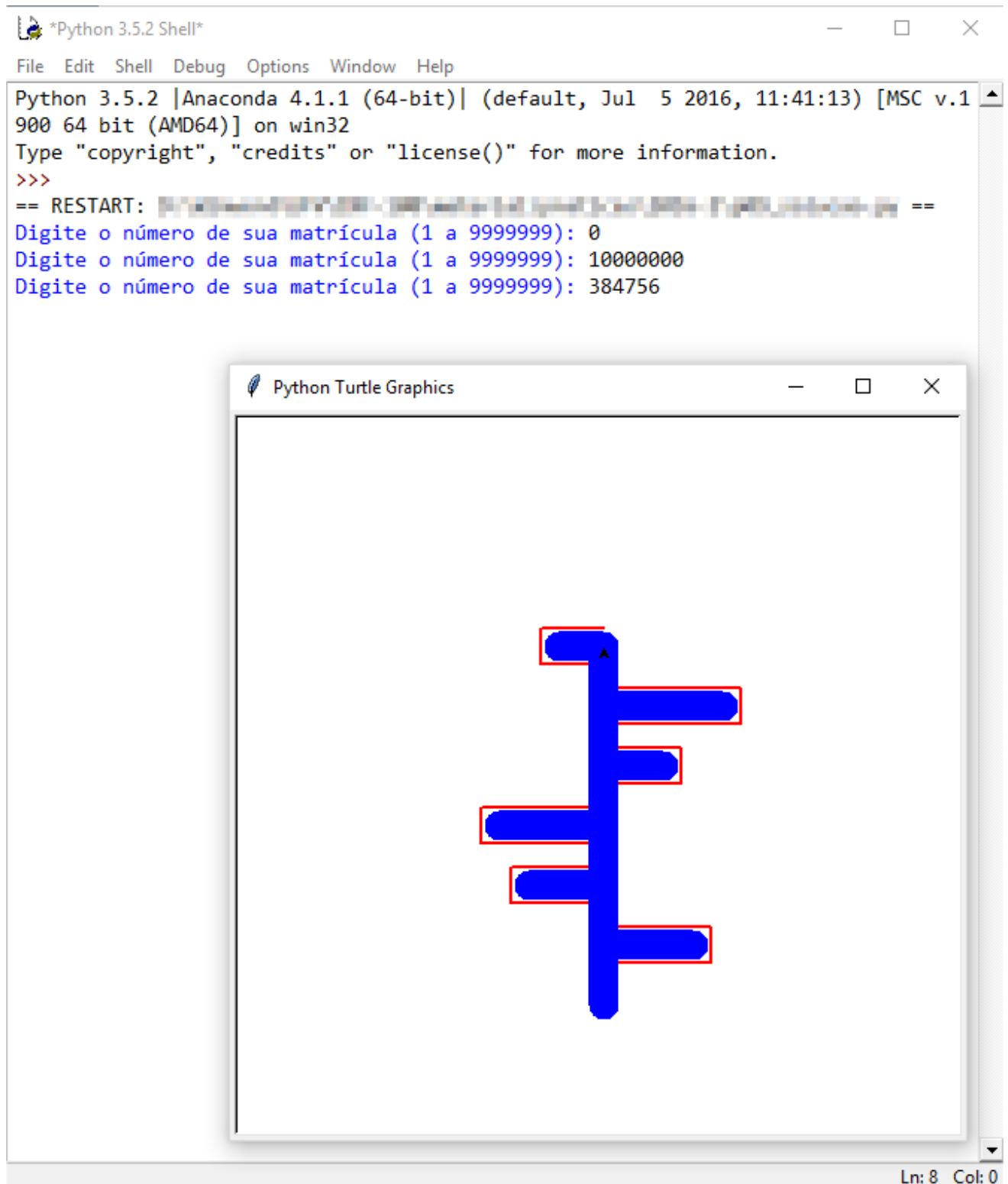
Para obter cada dígito d sucessivamente, basta fazer a divisão inteira do número de matrícula por 10. Por exemplo, se dividirmos 76543 por 10, teremos o quociente 7654 e o resto 3. Dessa forma, se tivermos o número (76543) dentro da variável `matricula`, por exemplo, então `matricula % 10` resultará em 3, e `matricula // 10` resultará em 7654. Se repetirmos essa divisão até que o quociente seja zero, pegaremos cada dígito sucessivamente, da direita para a esquerda.

Para ter certeza de que você desenhou a chave corretamente, seu programa chamará uma função já pronta que desenhará uma “fechadura” correspondente à sua chave. A chave deve se encaixar perfeitamente na fechadura, como mostram os exemplos abaixo:



Obs.: seu programa deverá funcionar para qualquer número de matrícula entre 1 e 9.999.999. O programa só deve desenhar a fechadura e a chave depois que o usuário digitar um valor dentro desse intervalo. Veja os comentários dentro do programa **p07.py** para mais detalhes.

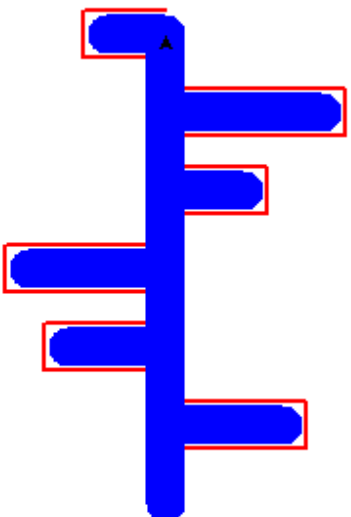
Segue abaixo um exemplo de execução do programa:



```

Python 3.5.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.5.2 [Anaconda 4.1.1 (64-bit)] (default, Jul  5 2016, 11:41:13) [MSC v.1
900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: ==
Digite o número de sua matrícula (1 a 9999999): 0
Digite o número de sua matrícula (1 a 9999999): 10000000
Digite o número de sua matrícula (1 a 9999999): 384756
  
```

Python Turtle Graphics



Ln: 8 Col: 0

☞ Não esqueça de preencher o cabeçalho com seus dados e uma breve descrição do programa.

Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p07.py**) através do sistema do LBI.