

Nome do arquivo a ser entregue: **p10.py**

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

As bibliotecas padrão de Python fornecem algumas ferramentas para trabalhar com **números (pseudo) aleatórios**:

- **random.randint(a, b)**: essa função retorna um valor inteiro aleatório N tal que $a \leq N \leq b$.
- **random.seed(a)**: essa função inicializa o gerador de números aleatórios, recebendo como parâmetro um valor inteiro que é usado como “semente”. Se o parâmetro **a** for omitido, a função usa a hora atual do computador como “semente”.

Se você estiver conduzindo experimentos usando valores aleatórios, pode ser importante ter que usar a mesma sequência de números vez após vez. Nesse caso, você pode usar **random.seed(a)** passando uma “semente” constante para sempre gerar a mesma sequência de valores. Por exemplo: **random.seed(0)**. Por outro lado, se você quiser gerar sempre sequências diferentes (como acontece em jogos, por exemplo), você pode usar “sementes” diferentes em cada execução do programa. Nesse caso, é só chamar **random.seed()**, omitindo o parâmetro.

Segue exemplo de um pequeno programa usando essas funções:

```
import random

random.seed()

print('Sequência de 10 números aleatórios entre 1 e 100:')
for i in range( 0, 10 ):
    n = random.randint( 1, 100 )
    print( n, end=' ')
print()
```

Nesta prática você deverá implementar o jogo do “adivinha o número”. Seu programa deverá efetuar as seguintes tarefas:

1. Pedir ao usuário para fornecer um número inteiro que funcionará como limite superior. O programa só deve permitir a entrada de valores maiores que zero.
2. Sortear um valor inteiro entre zero e o limite superior fornecido.
3. Exibir na tela uma mensagem confirmando o sorteio do número (veja os exemplos adiante).
4. Pedir ao usuário para tentar adivinhar o número sorteado.
5. Verificar se o usuário acertou o palpite. Caso não tenha acertado, informar se o número sorteado é maior ou menor do que o palpite e retornar ao passo 4. Caso tenha acertado, seguir para o passo 6.
6. Avisar que o usuário acertou o número sorteado, informar quantas tentativas foram necessárias e finalizar o programa.

Os passos 2 e 3 já estão prontos dentro do “esqueleto” do programa fornecido no sistema de entrega.

Seguem dois exemplos das “telas” de execução desse programa. As entradas de dados do usuário (pelo teclado) estão **destacadas**.

Exemplo 1:

```
Jogo 'Adivinhe meu Número'

Entre com o limite superior para o sorteio (> 0): -10
Entre com o limite superior para o sorteio (> 0): 0
Entre com o limite superior para o sorteio (> 0): 10
Acabei de sortear um número entre 0 e 10. Tente adivinhá-lo:

--> 5
Menor que 5.

--> 2
Você acertou em 2 tentativa(s)!
```

Exemplo 2:

```
Jogo 'Adivinhe meu Número'

Entre com o limite superior para o sorteio (> 0): 100
Acabei de sortear um número entre 0 e 100. Tente adivinhá-lo:

--> 50
Menor que 50.

--> 25
Maior que 25.

--> 35
Maior que 35.

--> 40
Maior que 40.

--> 44
Você acertou em 5 tentativa(s)!
```

Obs.: o número sorteado será aleatório e, portanto, poderá ser diferente daqueles mostrados nos exemplos acima.

☞ 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 (**p10.py**) através do sistema do LBI.