2.5

```
>>> help("random.randrange")
Help on method randrange in random:

random.randrange = randrange(start, stop=None, step=1, _int=<class 'int'>) method of random
.Random instance
    Choose a random item from range(start, stop[, step]).

    This fixes the problem with randint() which includes the
    endpoint; in Python this is usually not what you want.
```

The `randrange` function has three integer parameters: `start`, `stop`, and `step`. The function allows Python to return a random selected integer from the specified range, including the start point, but exclude the end point.

- The default `start` parameter is `0` when there is no specified `start` parameter.
- The default `step` parameter is `1` when there is no specified `step` parameter.
- One should always specify a `stop` parameter other than `0`. The actual endpoint of the range is (`stop-1`), which means that the function returns any integer from `start` to (`stop-1`).

Three ways to use the `randrange` function:

1. `random.randrange(stop)`
   Ex. `random.randrange(5)` returns any number among 0, 1, 2, 3, 4
2. `random.randrange(start, stop)`
   Ex. `random.randrange(1,5)` returns any number among 1, 2, 3, 4
3. `random.randrange(start, stop, step)`
   Ex. `random.randrange(3, 8, 2)` returns any number among 3, 5, 7

```
>>> help("random.randint")
Help on method randint in random:

random.randint = randint(a, b) method of random.Random instance
    Return random integer in range [a, b], including both end points.
```

The `randint` function returns a random integer N such that a <= N <= b. returns the same result as `randrange(a, b+1)`.

Ex. `random.randint(1,5)` returns any integer among 1, 2, 3, 4, 5
(`random.randrange(1,6)` returns any integer among 1, 2, 3, 4, 5)

2.11

```
>>> def average(N):
        Acc=0
        for x in range(1,N+1):
                Acc = Acc + x
                averageAcc = Acc/N
        return averageAcc

>>> average(1)
1.0
>>> average(2)
1.5
>>> average(3)
2.0
```

2.23

```
>>> not 7 > 3
False
```

2.27

```
>>> import random
>>> count = random.randint(1,10)
>>> count <= 10 and count >= 1
True
```

2.29

```
>>> if result==100:
        answer=1
else:
        answer=2
```

We supposed to receive an error message: `NameError: name 'result' is not defined`. So to test if the selection statement works,

```
>>> def random(result):
        if result==100:
                answer=1
        else:
                answer=2
        return answer

>>> random(80)
2
>>> random(100)
1
```

2.35

```
>>> def salary(rate, hours):
        if hours > 40:
                pay = rate*40 + 1.5*rate*(hours-40)
        else:
                pay = rate*hours
        return pay
```

EOC 2.8

```
>>> def function(n,i):
        x = 1
        for k in range(i):
                nextterm = (x+n/x)/2
                x = nextterm
        return nextterm
```

To test if the function works,

```
>>> function(4,4)
2.0000000929222947
>>> function(4,5)
2.000000000000002
>>> function(4,6)
2.0
```