Data structure Lab

Classes (Examples)

Dr. Halah H. Mahmoud

Computer science department / College of Science for Women

Example 1:

Write a Python class to implement pow(x, n).

- The following program compute power no of two integer numbers

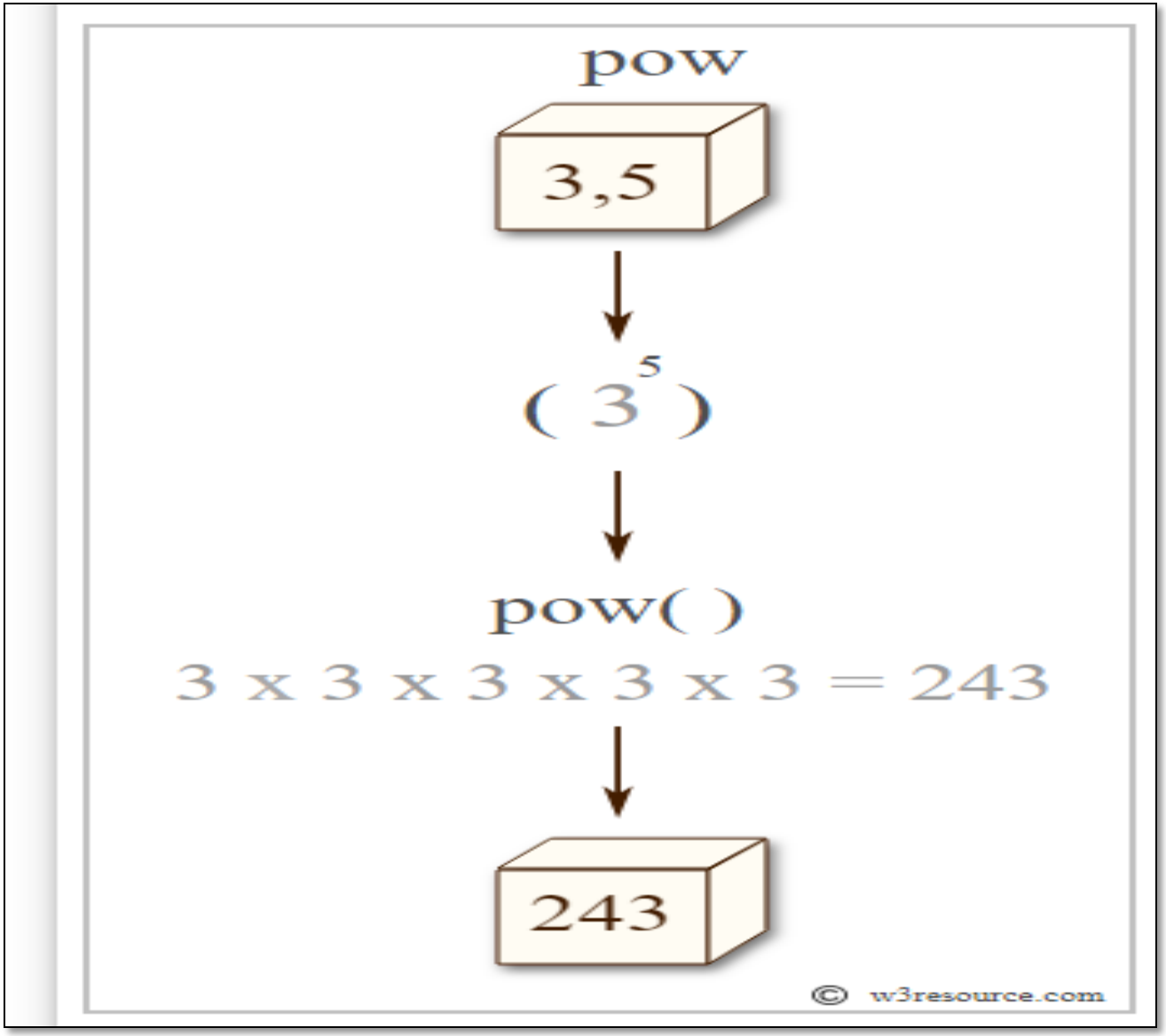
- X may be 0 or negative
- N may be 0 , more ore less 0.

Condition

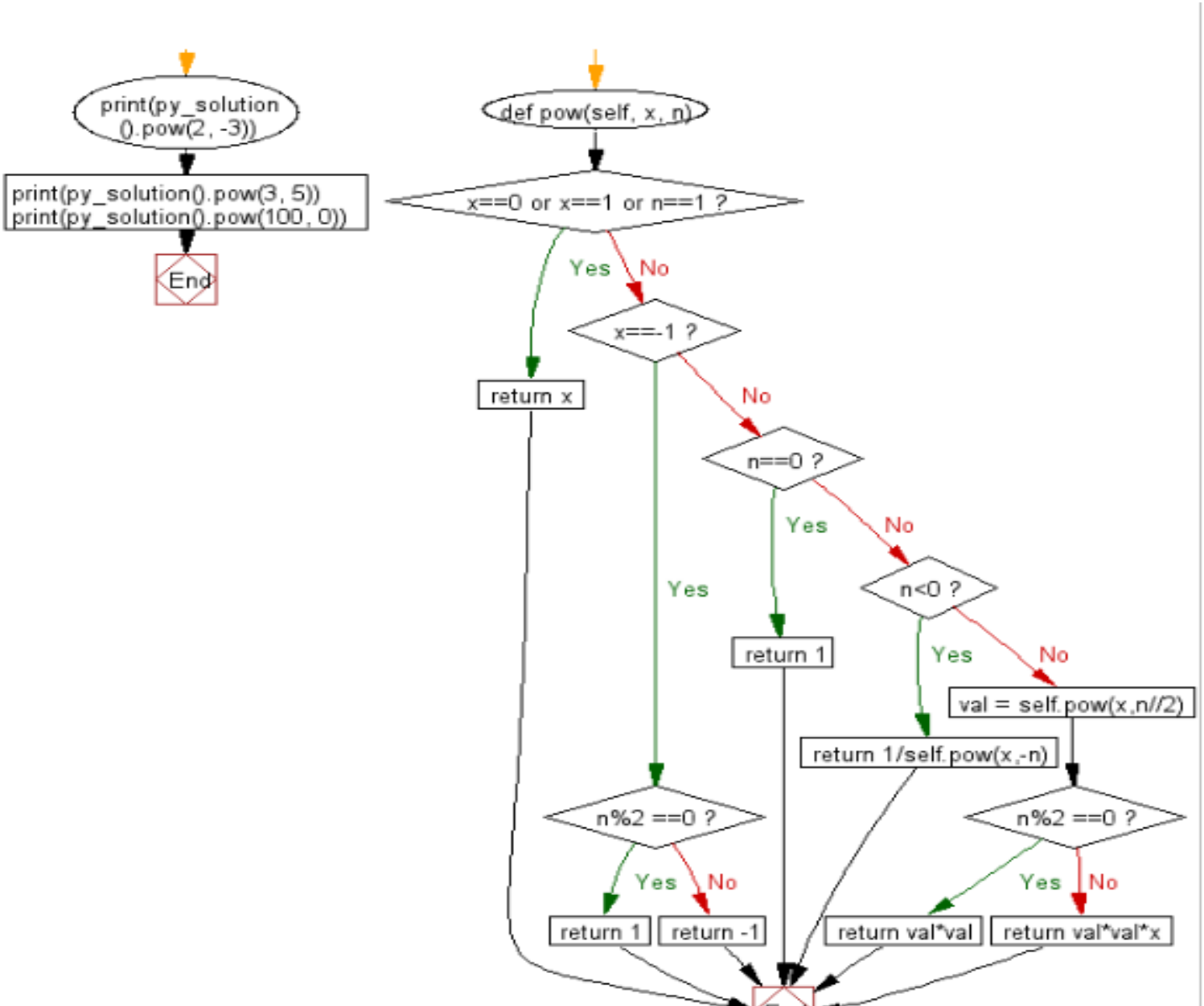
Output

0.125
243
1

Numbers enter must be
(Integer,, Positive or Negative)


pow
3,5
(3 5)
pow( )
3 x 3 x 3 x 3 x 3 = 243
243
© w3resource.com

FLOWCHART


print(py_solution
().pow(2, -3))
print(py_solution().pow(3, 5))
print(py_solution().pow(100, 0))
End
def pow(self, x, n)
x==0 or x==1 or n==1 ?
Yes
No
x==-1 ?
return x
No
n==0 ?
Yes
No
n<0 ?
Yes
return 1
Yes
No
val = self.pow(x,n//2)
return 1/self.pow(x,-n)
n%2 ==0 ?
n%2 ==0 ?
Yes
No
Yes
No
return 1
return -1
return val*val
return val*val*x

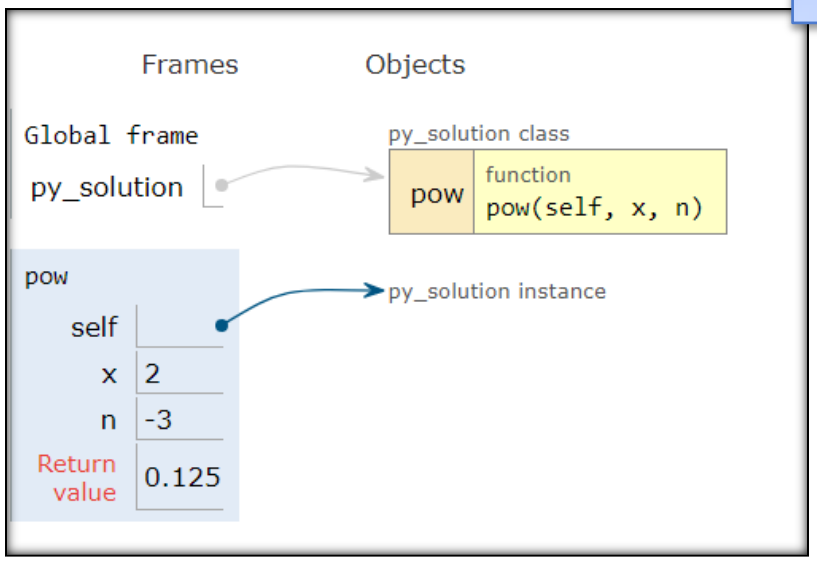
PROGRAM

```
class py_solution:
  def pow(self, x, n):
      if x==0 or x==1 or n==1:
          return x

      if x==-1:
          if n%2 ==0:
              return 1
          else:
              return -1
      if n==0:
          return 1
      if n<0:
          return 1/self.pow(x,-n)
      val = self.pow(x,n//2)
      if n%2 ==0:
          return val*val
      return val*val*x
print(py_solution().pow(2, -3));
print(py_solution().pow(3, 5));
print(py_solution().pow(100, 0));
```

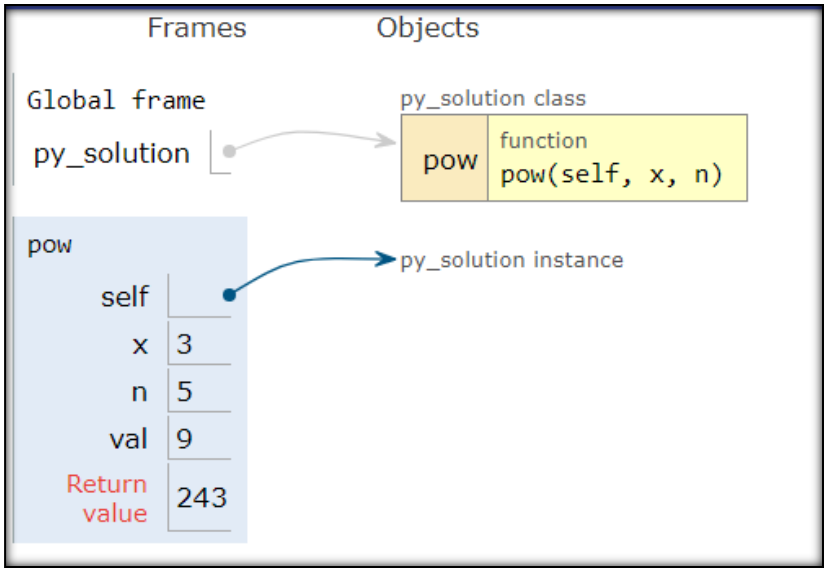
```
0.125
243
1
```

TRACING

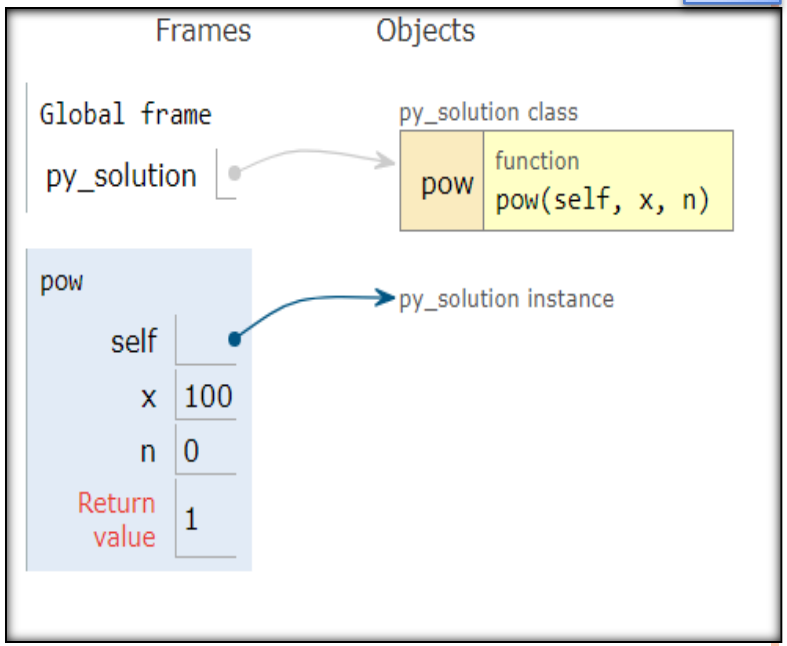
1


Frames
Objects
Global frame
py_solution
py_solution class
pow
function
pow(self, x, n)
pow
self
x 2
n -3
Return value 0.125
py_solution instance

2


Frames
Objects
Global frame
py_solution
py_solution class
pow
function
pow(self, x, n)
pow
self
x 3
n 5
val 9
Return value 243
py_solution instance

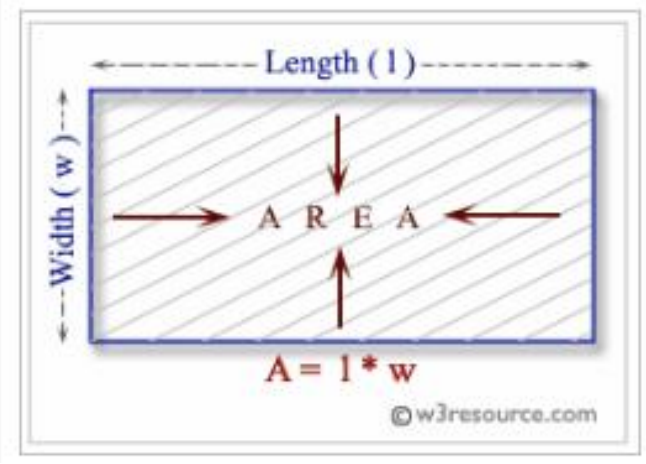
3


Frames
Objects
Global frame
py_solution
py_solution class
pow
function
pow(self, x, n)
pow
self
x 100
n 0
Return value 1
py_solution instance

EXAMPLE 2:

A class constructed by a length and width and a method which will compute the area of a rectangle

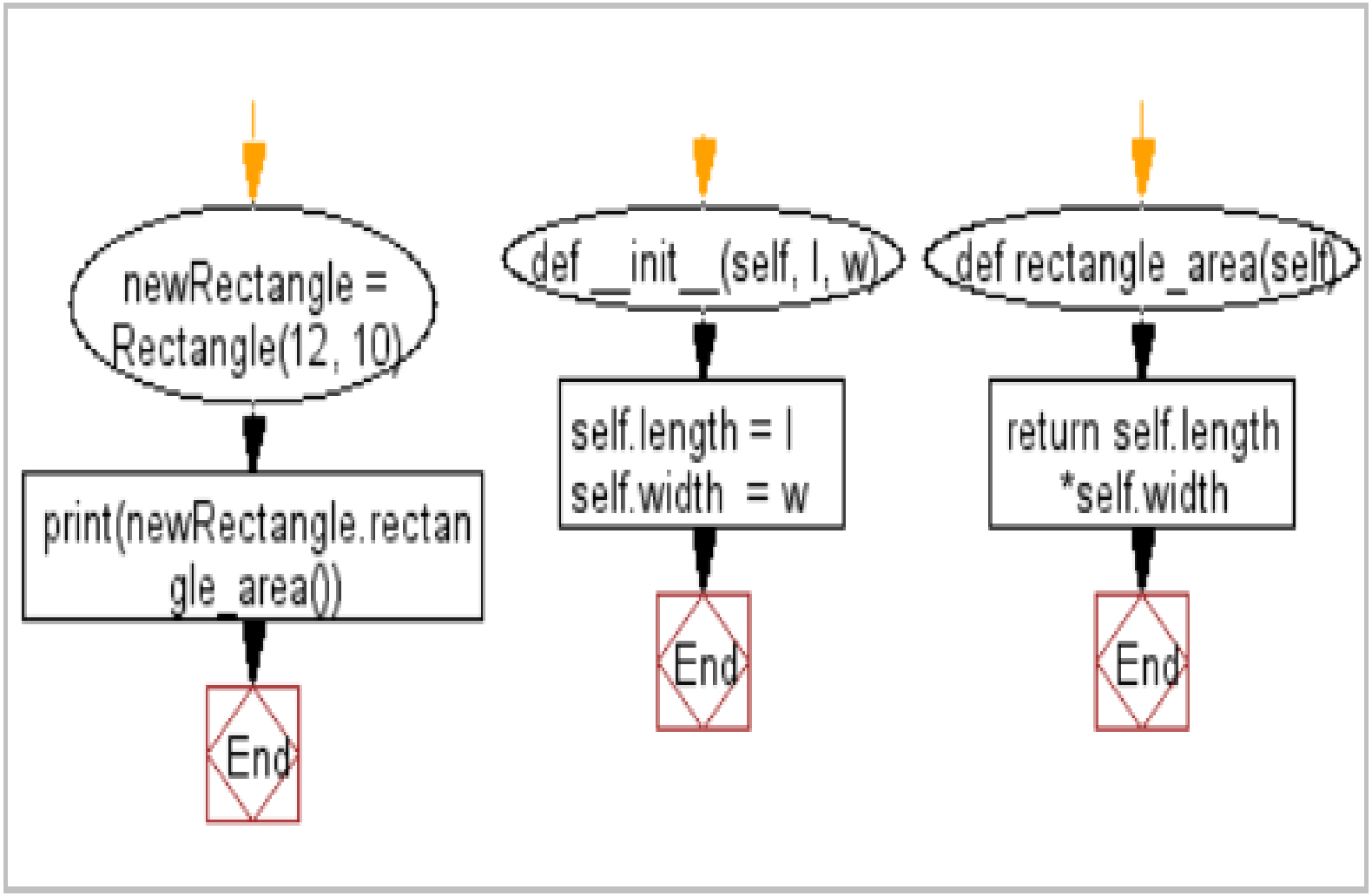
- In Euclidean plane geometry, a rectangle is a quadrilateral with four right angles. To find the area of a rectangle, multiply the length by the width.
  A rectangle with four sides of equal length is a square.
  Following image represents the area of a rectangle.



Output

120

FLOWCHART

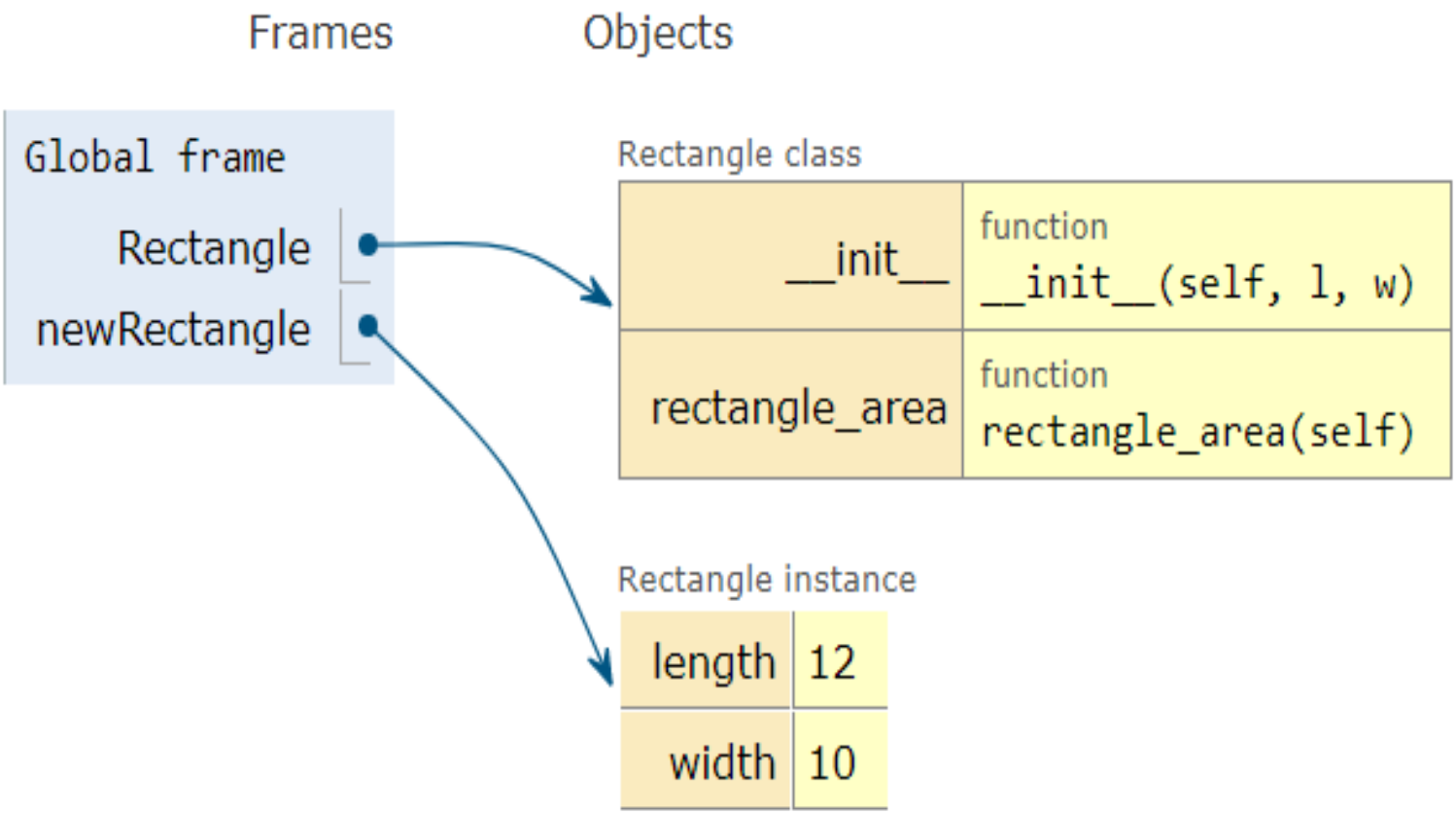

newRectangle = Rectangle(12, 10)
print(newRectangle.rectangle_area())
End
def __init__(self, l, w)
self.length = l
self.width = w
End
def rectangle_area(self)
return self.length *self.width
End

PROGRAM

```
class Rectangle():
    def __init__(self, l, w):
        self.length = l
        self.width  = w

    def rectangle_area(self):
        return self.length*self.width

newRectangle = Rectangle(12, 10)
print(newRectangle.rectangle_area())
```

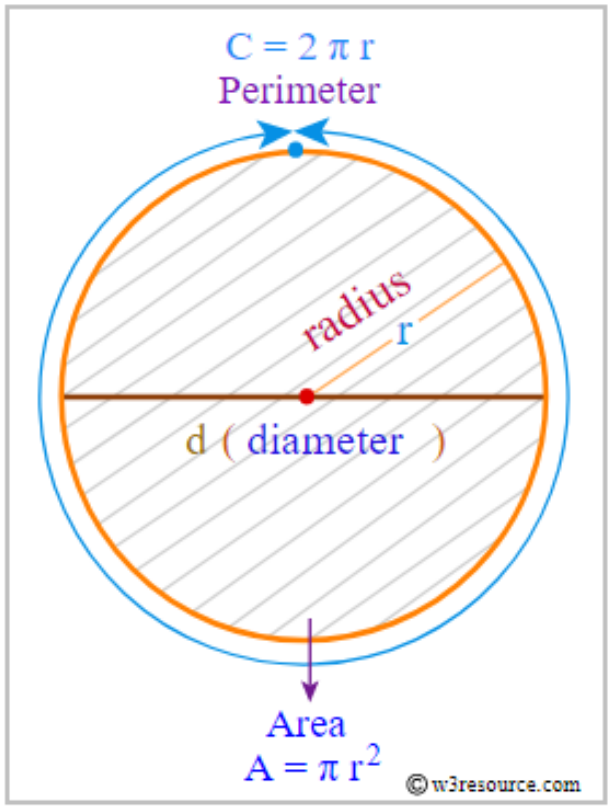
120

Tracing


Frames
Objects
Global frame
Rectangle
newRectangle
Rectangle class
__init__
function
__init__(self, l, w)
rectangle_area
function
rectangle_area(self)
Rectangle instance
length
12
width
10

Example 3:

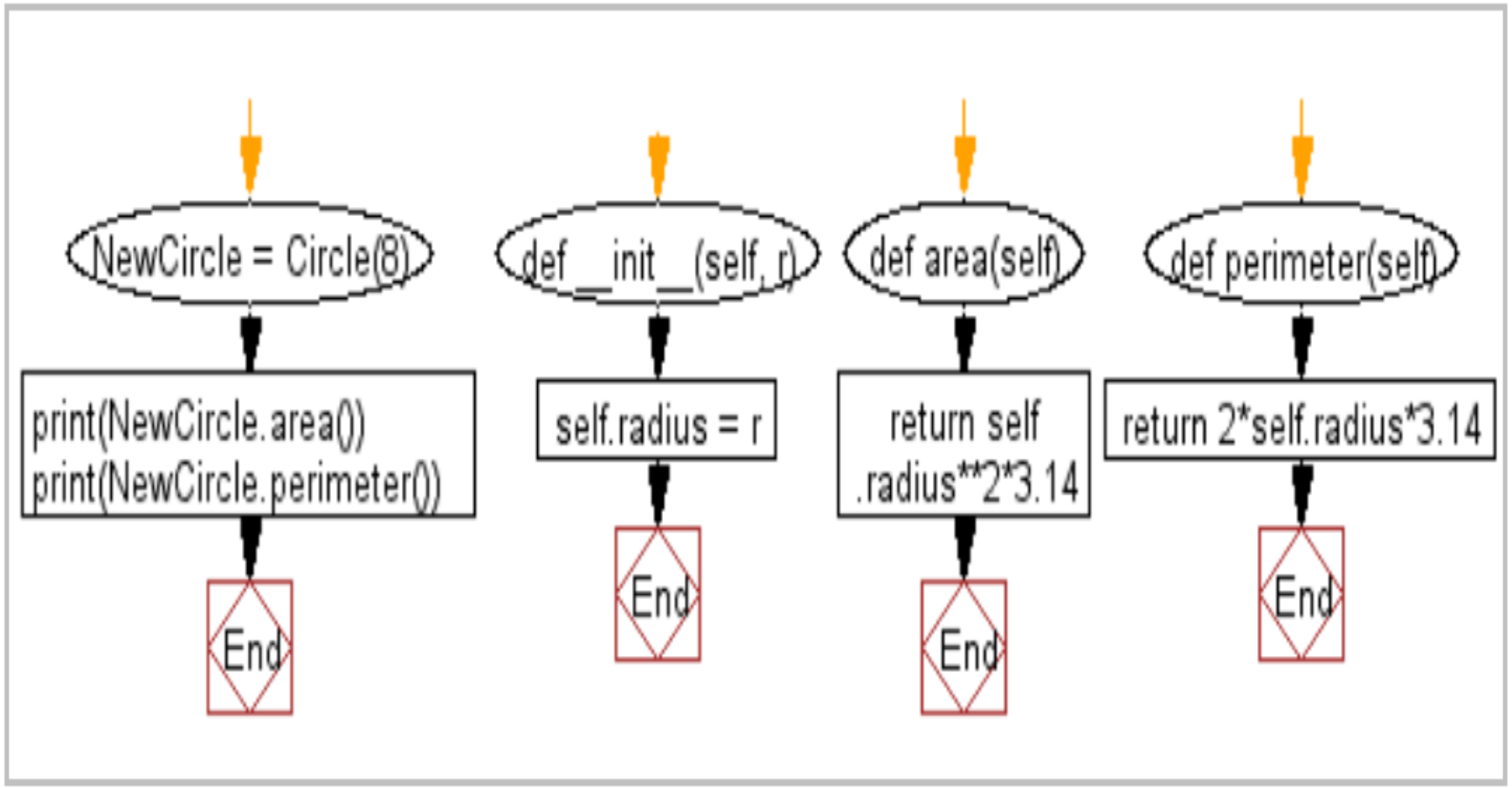
A class constructed by a radius and two methods which will compute the area and the perimeter of a circle



Output

200.96
50.24

FLOWCHART

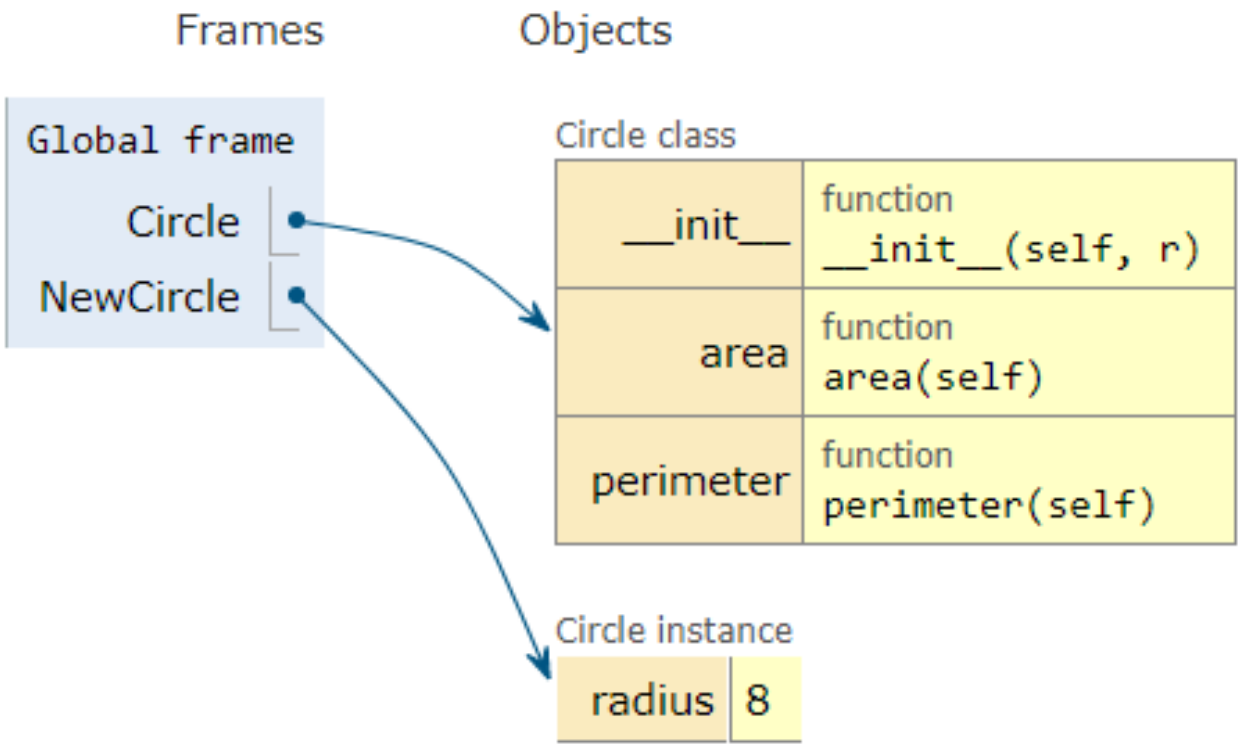

NewCircle = Circle(8)
print(NewCircle.area())
print(NewCircle.perimeter())
End
def __init__(self, r)
self.radius = r
End
def area(self)
return self
.radius**2*3.14
End
def perimeter(self)
return 2*self.radius*3.14
End

PROGRAM

```
class Circle():
    def __init__(self, r):
        self.radius = r

    def area(self):
        return self.radius**2*3.14

    def perimeter(self):
        return 2*self.radius*3.14

NewCircle = Circle(8)
print(NewCircle.area())
print(NewCircle.perimeter())
```

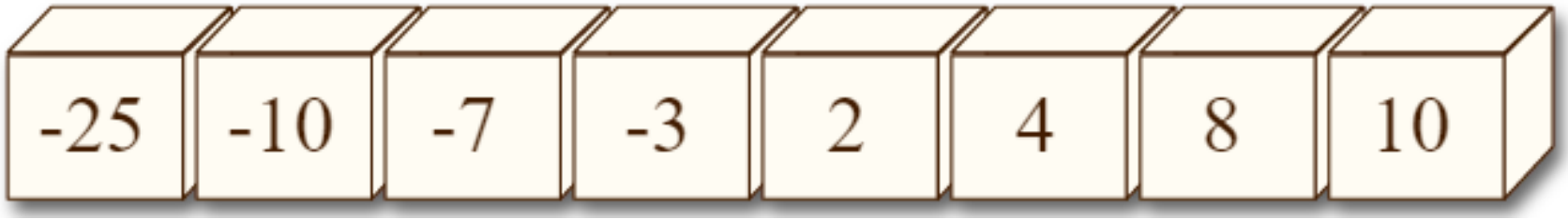
```
200.96
50.24
```

Tracing


Frames
Objects
Global frame
Circle
NewCircle
Circle class
__init__
function
__init__(self, r)
area
function
area(self)
perimeter
function
perimeter(self)
Circle instance
radius
8

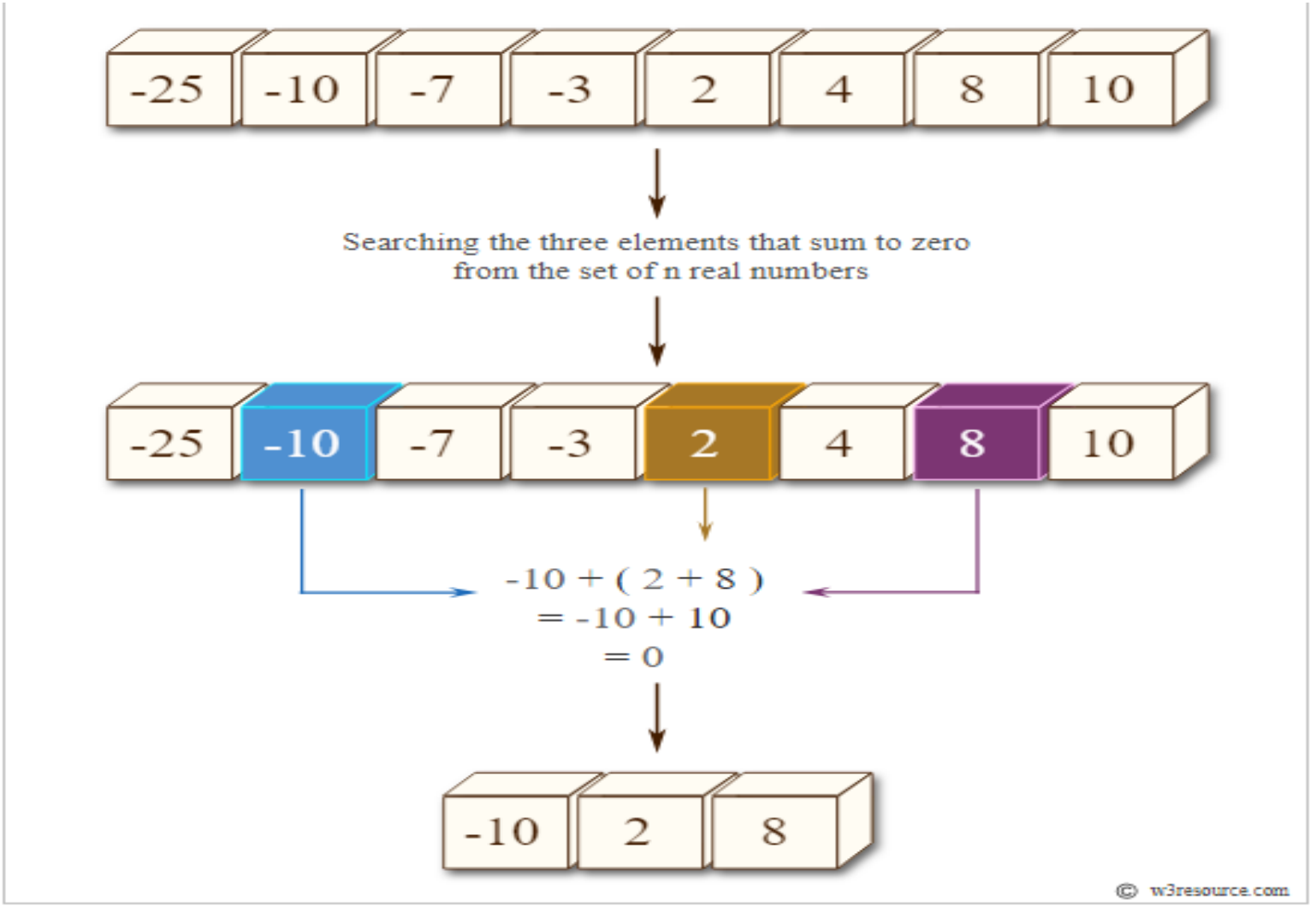
EXAMPLE 4:

Write a Python class to find the three elements that sum to zero from a set (array) of n real numbers.

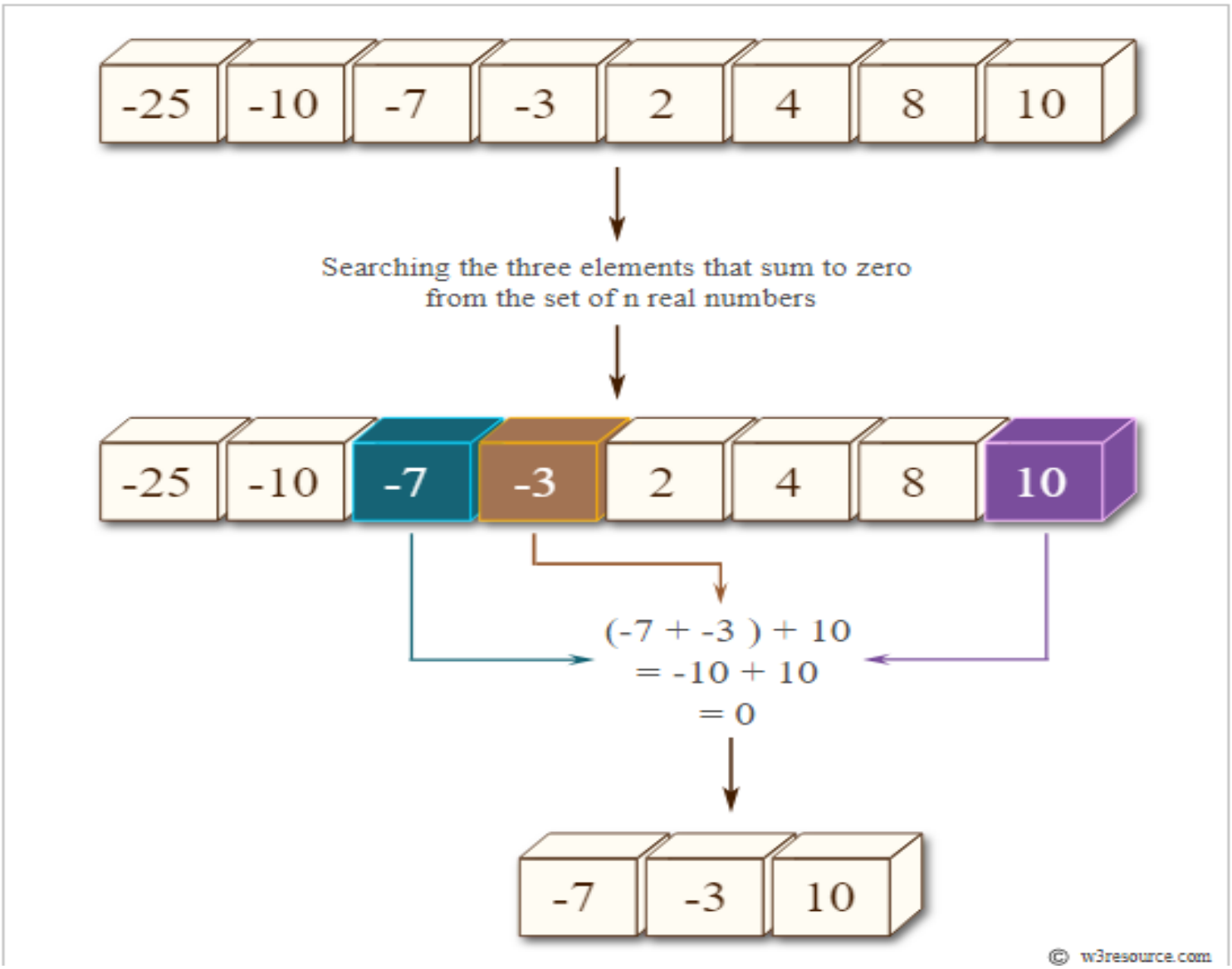


Output

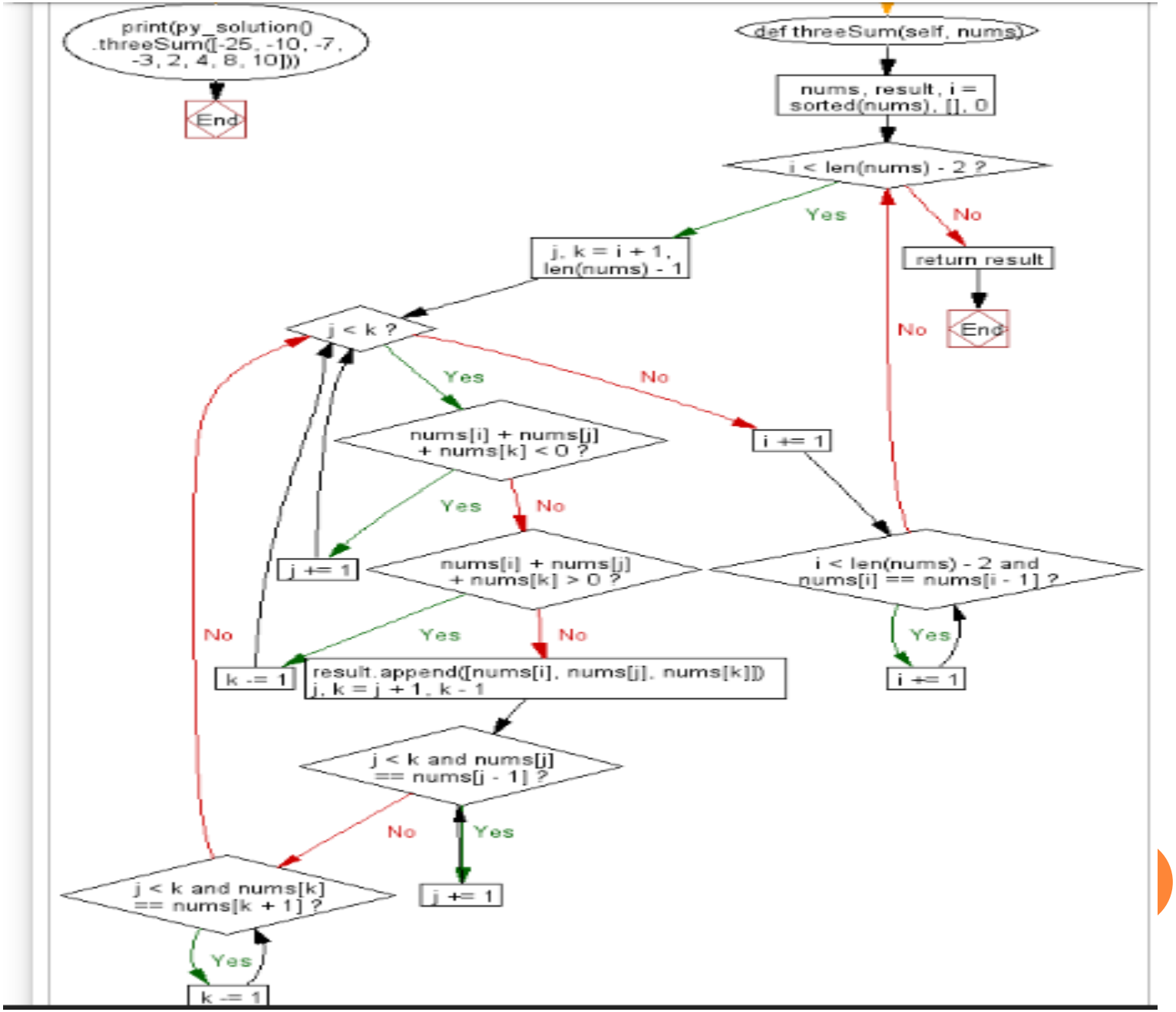
[[-10, 2, 8], [-7, -3, 10]]


-25 -10 -7 -3 2 4 8 10
Searching the three elements that sum to zero
from the set of n real numbers
-25 -10 -7 -3 2 4 8 10
-10 + (2 + 8)
= -10 + 10
= 0
-10 2 8
© w3resource.com


-25 -10 -7 -3 2 4 8 10


-25
-10
-7
-3
2
4
8
10
Searching the three elements that sum to zero
from the set of n real numbers
-25
-10
-7
-3
2
4
8
10
(-7 + -3) + 10
= -10 + 10
= 0
-7
-3
10
© w3resource.com

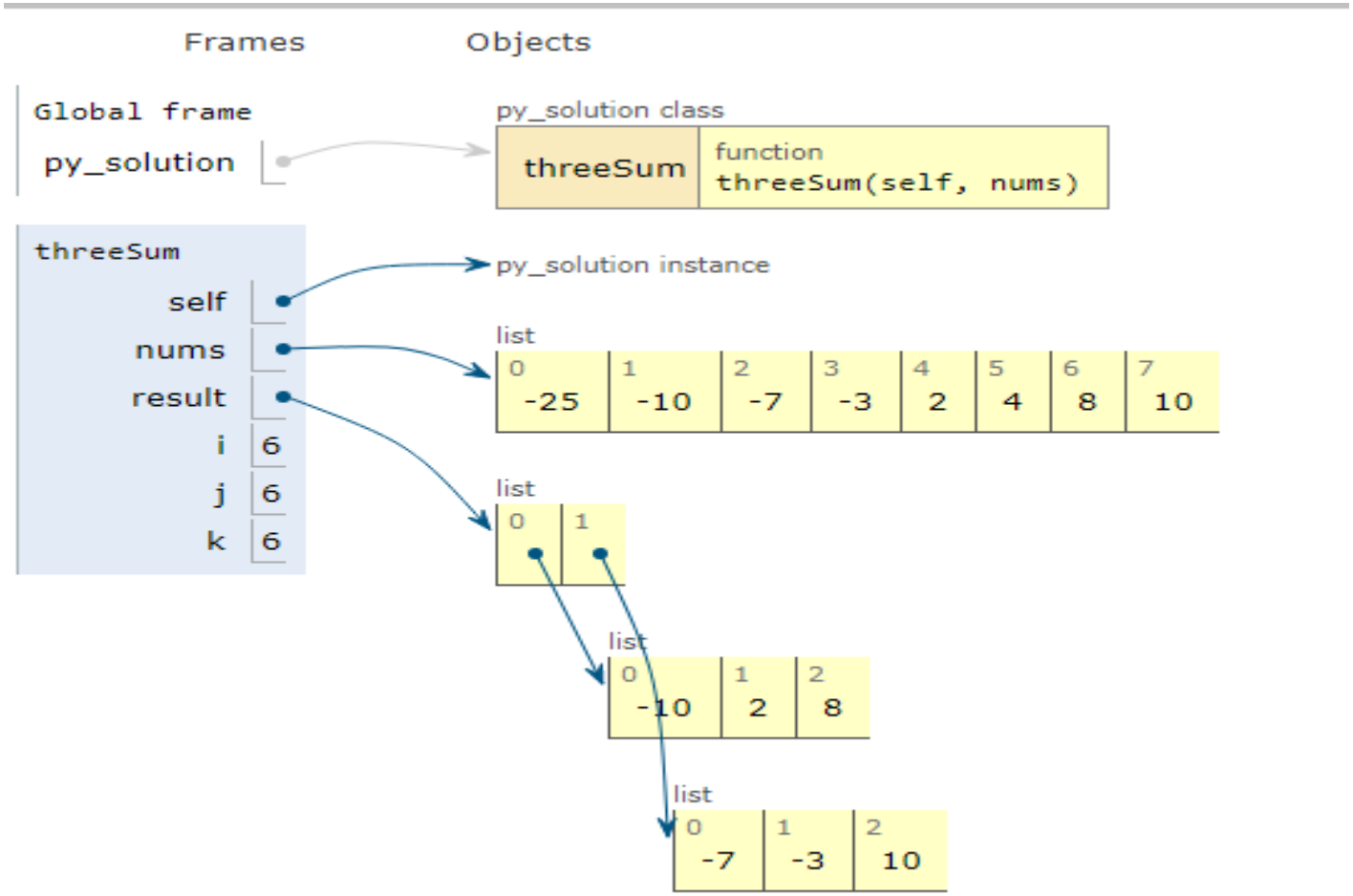
FLOWCHART


print(py_solution()
.threeSum([-25, -10, -7,
-3, 2, 4, 8, 10]))
End
def threeSum(self, nums)
nums, result, i =
sorted(nums), [], 0
i < len(nums) - 2 ?
Yes
No
j, k = i + 1,
len(nums) - 1
return result
End
No
j < k ?
Yes
No
nums[i] + nums[j]
+ nums[k] < 0 ?
i += 1
Yes
No
j += 1
nums[i] + nums[j]
+ nums[k] > 0 ?
i < len(nums) - 2 and
nums[i] == nums[i - 1] ?
No
Yes
No
Yes
k -= 1
result.append([nums[i], nums[j], nums[k]])
j, k = j + 1, k - 1
i += 1
j < k and nums[j]
== nums[j - 1] ?
No
Yes
j < k and nums[k]
== nums[k + 1] ?
j += 1
Yes
k -= 1

PROGRAM

```
class py_solution:
 def threeSum(self, nums):
        nums, result, i = sorted(nums), [], 0
        while i < len(nums) - 2:
            j, k = i + 1, len(nums) - 1
            while j < k:
                if nums[i] + nums[j] + nums[k] < 0:
                    j += 1
                elif nums[i] + nums[j] + nums[k] > 0:
                    k -= 1
                else:
                    result.append([nums[i], nums[j], nums[k]])
                    j, k = j + 1, k - 1
                    while j < k and nums[j] == nums[j - 1]:
                        j += 1
                    while j < k and nums[k] == nums[k + 1]:
                        k -= 1
            i += 1
            while i < len(nums) - 2 and nums[i] == nums[i - 1]:
                i += 1
        return result
print(py_solution().threeSum([-25, -10, -7, -3, 2, 4, 8, 10]))
```

TRACING


Frames
Objects
Global frame
py_solution
py_solution class
threeSum
function
threeSum(self, nums)
threeSum
self
nums
result
i 6
j 6
k 6
py_solution instance
list
0 -25
1 -10
2 -7
3 -3
4 2
5 4
6 8
7 10
list
0
1
list
0 -10
1 2
2 8
list
0 -7
1 -3
2 10

Any Questions