WWPP: What Would Python Print?

Siva.Jasthi@gmail.com

Learn and Help: Python 101

www.learnandhelp.com

A. Variables, Input and Output

```
#WWPP A.1
a = 10
b = 20
a = b
print(a, b)
```

Show hidden output

```
#WWPP A.2
x, y, z = 1, 2, 3
print(x + y + z)
```

Show hidden output

```
#WWPP A.3
x, y, z = 1, 2, 3
print(x, y, z, sep='+')
```

Show hidden output

```
#WWPP A.4
x, y, z = 1, 2, 3
print(x, y, z, end='+')
```

Show hidden output

```
#WWPP A.5
X = 2
Y = 5
Z = 3
X = X + 2
Y = X + 2
Z = X + 2
print(X, Y, Z)

```

Show hidden output

```
#WWPP A.6
X = 25
Y = 50
Z = 75

X = Y
Y = Z
Z = X
print(X,Y,Z)

# Which of the variables have the value equal to 50
```

Show hidden output

```
#WWPP A.7
A = 10
B = 20
C = 30
D = 40
X = 20

B = X + B
A = X + 1
D = C + D - 2
print(A, B, C, D)
```

Show hidden output

```
#WWPP A.8
first = True
second = False

second = first
first = second
print(first, second)

```

Show hidden output

```
#WWPP A.9
x = 1
y = 2
z = x
x = y
y = z
print(x+y)
```

Show hidden output

```
#WWPP A.10
x, y = 10, 20
x, y = y, x
print(x - y)
```

Show hidden output

```
#WWPP A.11
a = b = c = 2
x = a + b + 3 * c
print(x)
```

Show hidden output

```
#WWPP A.12
print(1,2,3, end='4', sep='')
```

Show hidden output

```
#WWPP A.13
print(1, 3, sep='2', end='4')
```

Show hidden output

```
# @title
#WWPP A.14 (TBD control characters)
x = '\\nit'
print(x)
```

Show hidden output

B. Operators and Data Types

```
 #WWPP B.1
x = 2 + 3 * 4 / 2
print(x)
```

Show hidden output

```
#WWPP B.2
x = 2
x += 1
y = 2
y *= 2
y **= 2
print(x+y)

```

Show hidden output

```
#WWPP B.3
x = 10 // 3
y = 10 % 3
print(x + y)
```

```
#WWPP B.4
a = 5 + 2 ** 2
a = a ** 0.5
print(a)
```

Show hidden output

```
#WWPP B.5
a = 5 / 2 * 2
a += 2
print(a)
```

Show hidden output

```
#WWPP B.6
a = 3
b = -a * 2
x = -a - (b + 2) * a
print(x)
```

Show hidden output

```
#WWPP B.7
a = 6
b = 3
c = 6.0
d = 3.

x = a // b
y = a // d
z = c // b

print(x + y + z)
```

Show hidden output

```
#WWPP B.8
a = -7
b = 3
x = a // b
print(x)
```

Show hidden output

```
#WWPP B.9
x = 10 % 4
y = 10.0 % 3
z = 10 % 4.5

print(x + y + z)
```

Show hidden output

```
#WWPP B.10
x = 2 ** 3 ** 4
y = (2 ** 3) ** 4
z = 2 ** (3 ** 4)

a = x == y
b = x == z

print(a, b)
```

Show hidden output

```
#@title
#WWPP B.11
x = 2 * 3 // 4
y = (2 * 3) // 4
z = 2 * (3 // 4)

a = x == y
b = x == z

print(a and b)
```

Show hidden output

```
#WWPP B.12
x = 9 % 6 % 2
y = 2 ** 2 ** 0
z = 2 % 1
print(x + y + z)
```

Show hidden output

```
#WWPP B.13
a = 4
b = a + 1
```

```
c = 2 * b + 2
print(a+b+c)
```

Show hidden output

```
#WWPP B.14
a = '2'
x = a * 3
print(x)
```

C. Conditions and Boolean Logic

```
#WWPP C.1
a = True
b = False
c = True
if (a and b):
  print(10)
if (a or c):
  print(20)
if (b and c):
  print(30)


```

```
#WWPP C.2
a = True
b = False
c = True
if (a and b):
  print('apple')
elif (a and c):
  print('berry')
elif (b and c):
  print('cherry')
else:
  print('dragon')
```

Show hidden output

```
# WWPP C.3
a = True
b = True
c = False

```

```
x = (a and c) and (a and b)
print(x)
```

Show hidden output

```
# WWPP C.4
a = True
b = False
c = False

x = (a and c) or (a and b)
print(x)
```

Show hidden output

```
# WWPP C.5
a = 1 or 9 or 2
b = 1 and 9 and 2
c = 0 or 4 or 5
print(a + b + c)
```

Show hidden output

```
#WWPP C.6
num1 = 6
num2 = 4
num3 = 10

if (num1 < num2):
   num1 = num2
else:
   num3 = num2

if (num2 >= num3):
  num1 = num2 + num3

x = num1 + num2 + num3
print(x)
```

Show hidden output

```
#WWPP  C.7
score1 = 350
score2 = 210

result = 1
if (score1 > 500):
  result = result + 1
```

```
 8   if (score2 > 500):
 9     result = result + 1
10   else:
11     result = result + 2
12 else:
13   result = result + 5
14   if (score2 > 500):
15     result = result + 1
16   else:
17     result = result -1
18
19 print(result)
```

Show hidden output

```
 1 # WWPP C.8
 2 a, b, c = 10, 20, 5
 3
 4 x = (a != b)
 5 y = (c + b < a)
 6 z = (a > c)
 7
 8 p = (x and y)
 9 q = (y or z)
10
11 r = (p or q)
12
13 print(r)
```

Show hidden output

```
 1 #WWPP  C.9
 2 age = 20
 3
 4 if (age <= 1):
 5     x = 10
 6
 7 else:
 8     if (age > 1) and (age < 13):
 9         x = 20
10
11     else:
12         if age >= 13 and (age < 20):
13             x = 30
14
15         else:
```

```
            if (age >= 20) and (age < 100):
                x = 40

            else:
                x = 50

print(x)
```

Show hidden output

```
# WWPP C.10

list_1 = [2]
list_2 = []

x = 0
if list_1:
  x = x + 1
if list_2:
  x = x + 1
if list_1 and list_2:
  x = x + 1
if list_1 or list_2:
  x = x + 1

print(x)
```

Show hidden output

```
# WWCP C.11
a = [True, False, 1, 0]
x = a.count(True)
print(x)
```

Show hidden output

```
# WWCP C.12
a = [1, 2, 3]

if a:
  x = sum(a)
else:
  x = len(a)

print(x)

```

Show hidden output

D. Loops, iterations

```
# WWCP D.1
i = 0
while (i < 5):
    if i == 3:
        break
    i += 1
    print(i, end = ' ')

```

```
# WWCP D.2
i = 1
while (i <= 4):
    i += 1
    if i == 3:
        continue
    print(i, end = ' ')

```

Show hidden output

```
# WWCP D.3
x = range(1, -1, -1)
print(list(x))
y = range(1, -1, 1)
print(list(y))
z= range(10, 2, -2)
print(list(z))
```

Show hidden output

```
# WWCP D.4
while True:
  print(10, end = '|')
  while False:
    print(20, end = '|')
  break
print(30, end = '')
```

```
# WWPP D.5
a = 3
while (a < 10):
  print(a, end = ' ')
```

```
  a = a + 3

```

Show hidden output

```
# WWPP D.6
a = 3
while (a < 10):
  a = a + 3
  print(a, end = ' ')

```

Show hidden output

```
# WWPP D.7
a = 0
total = 0

while (total < 10):
  a = a + 3
  total = total + a

print(a)

```

Show hidden output

```
# WWPP  D.8
a = 0
total = 0

while (total < 10):
  a = a + 3
  total = total + a

print(total)

```

Show hidden output

```
# WWPP D.9
a = 0
total = 0

while (a < 10):
  a = a + 3
  total = total + a

```

```
print(total)

```

Show hidden output

```
# WWPP D.10
a = 0
total = 0

while (a < 10):
  total = total + a
  a = a + 3

print(total)
```

Show hidden output

```
# WWPP D.11
x = 0
while (x <= 200):
  x += 2

print(x)
```

Show hidden output

```
# WWPP D.12
for i in range(1, 11):
  if i % 2 == 0:
    x = i / 2
    y = i // 2

print(y, x, sep = '')
```

```
55.0
```

```
# WWPP D.13
for i in range(1, 11):
  if i % 2 == 1:
    x = i // 2

print(x)
```

Show hidden output

```
# WWPP D.14
total = 10
for i in range(1, 11, 3):
  total += i
```

```
5
6 print(total)
```

Show hidden output

```
1 # WWPP D.15
2 total = 0
3
4 for i in range(1, 11, 3):
5   if i % 2 == 0:
6     continue
7   else:
8     total += i * 2
9
10 print(total)
```

Show hidden output

```
1 # WWPP D.16
2 for i in range(4):
3   for j in range(3):
4     result = i * j
5
6 print(result)
```

Show hidden output

```
1 # WWPP D.17
2 result = 0
3
4 for i in range(10):
5   for j in range(-1, -10, -1):
6     result += 1
7
8 print(result)
```

Show hidden output

```
#WWPP D.18
var = 10
for i in range(10):
  for j in range(2, 10, 1):
    if var % 2 == 0:
      continue
  var += 1

print(var)

```

Show hidden output

```
#WWPP D.19
x = sum(range(1,4)) + sum (range(-2, -5, -1)) + sum(range(-1,
print(x)
```

```
# WWPP D.20
result = 0
for c in "python":
  if c == 'o':
    break
  if c == 'h':
    continue
  result += 2

print(result)


```

Show hidden output

```
# WWPP D.21
for num in range(2, -2, -2):
  print(num, end = ' ')

```

Show hidden output

```
# WWPP D.22
nums = [10, 20, 30]
objs = ['Computer', 'Mouse', 'Desk']
for x in nums:
  for y in objs:
    z = str(x) + y

print(z)
```

Show hidden output

```
# WWPP D.23
for num in range(10, 14):
  for i in range(2, num):
    if num % i == 1:
      result = num
      break

print(result)

```

Show hidden output

```
#WWPP D.24
x = 4
while x < 6:
  x += 1
else:
  x += 2

print(x)
```

Show hidden output

```
#WWPP D.25
x = 4
for x in "python":
  ch = x
else:
  x += 'o'

print(x)
```

Show hidden output

```
#WWPP D.26
i = 0
while i <= 5:
  i += 1
  if i %2 == 0:
    break
  print('*')
else:
  print('*')
```

Show hidden output

```
#WWPP D.27
x = 5
y = 11
while True:
  x = x + 1
  y = y - 1
  if (x == y):
    break

print(x + y)
```

Show hidden output

```
#WWPP D.28
x = 5
y = 11
while True:
  if (x == y):
    continue
  x = x + 1
  y = y - 1


print(x + y)
```

Show hidden output

```
#WWPP D.29
import random
while True:
  x = random.randint(1, 10)
  if (2 ** x == 8):
    break

print(x)
```

Show hidden output

E. Functions

```
#WWPP E.1
def add(a, b):
  return a+3, b+4

result = add(1, 2)
print(result[1])
```

```
#WWPP E.2
num = 2
def fun_x(num = 4):
  return num + num

num = fun_x( )
print(num)
```

```
#WWPP E.3
num = 2
def fun_x(num):
  return num + num

num = fun_x(num)

result = fun_x(num)
print(num + result)
```

Show hidden output

```
#WWPP E.4
def print_name(name, marks):
  print(name)

print_name(2, "alex")
```

```
#WWPP E.5
def fun_y(num = 2):
  num = num + 3
  return num

x = fun_y(5)
print(x)

```

```
#WWPP E.6
def fun_y(num = 2):
  num = num + 3
  return num

x = fun_y()
print(x)

```

Show hidden output

```
#WWPP E.7
c = 4
def fun_y(a = 5, b = 5):
  num = a + b + c
  return num

x = fun_y()
print(x)

```

Show hidden output

```
#WWPP E.8
def fun_x(a = 1, b = 2):
  return (a + b)

x = fun_x(1,2)
y = fun_x(2)
z = fun_x()
print(x + y + z)
```

Show hidden output

```
#WWPP E.9
def fun_x(a, b = 5):
  return (a - b)

x = fun_x(4, 2)  # Positional
y = fun_x(b = 1, a = 3) # Keyword
z = fun_x(7) # Optional
print(x + y + z)
```

Show hidden output

```
#WWPP E.10
def get_max(a, b):
  return max(a,b)

x = get_max(get_max(5,6), get_max(7,8))
# x = get_max(6, get_max(7,8))
# x = get_max(6, 8)
# x = 8
print(x)
```

```
#WWPP E.11
def fun_x(input_list):
  a,b,*c = input_list
```

```
  return a,b,len(c)

x = fun_x([1,2,3,4,5])
print(sum(x))
```

Show hidden output

```
#WWPP E.12
x = 3
def fun_x(x):
  x = x + 3

fun_x(x)
print(x)
```

Show hidden output

```
#WWPP E.13
def fun_x(input_list):
  a = input_list[:-1]
  return len(a)


x = fun_x([1,2,3])
y = fun_x(['a', 'b', 'c', 'd'])
z = fun_x(list(range(5)))

print(x+y+z)
```

Show hidden output

```
#WWPP E.14
def foo():
  print(var+1, end='')

var = 1
foo()
print(var)

```

Show hidden output

```
#WWPP E.15
def foo(x):
  x = x + 10
  return x

a = 5
```

```
b = foo(a)
print(a + b)
```

Show hidden output

```
#WWPP E.16
a = 10
def foo( ):
  x = 20
  return x

# call foo()
b = foo()

print(a + b)

```

```
#WWPP E.17
numbers = [4,5]

def foo( ):
  numbers.append(3)
  numbers.append(2)


foo()
x = sum(numbers)
print(x)

```

Show hidden output

```
#WWPP E.18
numbers = [4,5]
def foo( ):
  numbers = []
  numbers.append(3)
  numbers.append(2)


foo()
x = sum(numbers)
print(x)
```

Show hidden output

```
#WWPP E.19
numbers = [4,5]
def foo( ):
  global numbers
  numbers = []
  numbers.append(3)
  numbers.append(2)


foo()
x = sum(numbers)
print(x)
```

Show hidden output

```
#WWPP E.20
def fun(x):
   global y
   y = x * x
   return y

a = fun(1)
b = fun(2)
print(a + b + y)
```

Show hidden output

```
#WWPP E.21
def foo(*nums):
  a = len(nums)
  b = sum(nums)
  return a,b

a, b = foo(-3, -2)
x = b // a
print(x)

```

Show hidden output

```
#WWPP E.22
def get_id_with_max_marks(**kwargs):
  max_marks = 0
  our_key = ''
  our_marks = 0
  for k,v in kwargs.items():
    if v > max_marks:
```

```
        our_key = k
        our_marks = v
        max_marks = our_marks
    return our_key, our_marks

# testing
x, y = get_id_with_max_marks( alex = 6, barb = 98, chris = 89,
print(x)
```

Show hidden output

F. Exceptions

```
#WWPP F.1
try:
    x = [1, 2, 3]
    print(x[3])
except:
    print(20)
```

Show hidden output

```
#WWPP F.2
try:
    x = [1, 2, 3]
    print(x[-4])
except IndexError:
    print(10, end='')
finally:
    print(1)
```

```
#WWPP F.3
try:
    x = "python"
    x = len(x)
except KeyError:
    x = 30
except IndexError:
    x = 40
finally:
  print(x)
```

```
#WWPP F.4
x = 10
try:
  x = 10/0
```

```
except:
  pass

print(x)
```

```
#WWPP F.5
x = 4
try:
  x = x // 2
except:
  print(x + 1, end='')
else:
  print(x + 2, end='')
finally:
  print(x + 3, end='')

print(x)
```

Show hidden output

G. Strings

```
# WWPP G.1

s1 = 'Ohio'
s2 = 'Texas'
s3 = 'Arizona'

s_list =  [s1, s2, s3]

x = max(s_list)
print(x)
```

Show hidden output

```
# WWPP G.2
s1 = 'OHIO\n'
s2 = 'TEXAS\n'

a = len(s1)
s2 = s2[:-1]
b = len(s2)

print(a+b)
```

Show hidden output

```
# WWPP G.3
s = 'python programming is cool'
a = s.split()
print(a[-2])

```

Show hidden output

```
#WWPP G.4
name = 'Robert Downy Junior'

name_list = name.split()

print(name_list[0][0] + name_list[1][0] + name_list[2][0])
```

```
#WWPP G.5
input_str = '101'
x = 0
for ch in input_str:
   x += int(ch) + 1
print(x)
```

Show hidden output

```
#WWPP G.6
vowels = 'aeiou'
if 'p' not in vowels:
  print(10)
else:
  print(20)
```

```
#WWPP G.7
name = 'Python'
x = 'C' + name[1::1]
print(x)
```

```
#@title
#WWPP G.8
str_1 = '''a
b
c
d'''
x = len(str_1)
print(x)
```

Show hidden output

```
#WWPP G.9
str_1 = 'alphabet'
x = min(str_1)
print(x)
```

Show hidden output

```
#WWPP G.10
str_1 = 'A zebra'
a = min(str_1)
x = str_1.index(a)
print(x)
```

Show hidden output

H. Lists

```
#WWPP H.1
my_list = [1, 2, 3, 3]
my_list.remove(3)
x = len(my_list)
y = sum(my_list)
print(x + y)
```

```
7
```

```
#WWPP H.2
my_list = [6, 7, 3, -4, 9]
x = my_list.pop()
y = my_list.pop()
print(x//y)
```

```
6
```

```
#WWPP H.3
x= [2] * 3
print(sum(x) + len(x))
```

```
4
5
```

```
#WWPP H.4
nums_1 = [1, 0, 3, 5, 2, 17, 19]
nums_2 = []
for elem in nums_1:
  if elem % 2 == 0:
    nums_2.append(elem)
print(sum(nums_2) + max(nums_2) + len(nums_2) + min(nums_2))

```

```
#WWPP H.5
x, y = [1, 3], [5, 4]
z = x + y
print(len(z))
```

```
#WWPP H.6
some_list = [1, 2, 3, 4, 5, 6, 7]
new_list =  [2 * x for x in some_list]
x = max(new_list) + min(new_list)
print(x)
```

```
#WWPP H.7
names = ['chris', 'barbara', 'anna']
x = max(names)
y = min(names)
print(len(x) - len(y))
```

```
#WWPP H.8
a, b, c = ['a', 'x', 'p', 'q', 'b', 'r'][1::2]
print(a+b)

```

```
#WWPP H.9
x = 10
y = -20
x, y = [y, x][::-1]
print(x-y)

```

Show hidden output

```
#WWPP H.10
my_list = [1, 2]

for v in range(2):
  my_list.insert(-1, my_list[v])
```

```
6
7 print(my_list)
```

Show hidden output

```
1 #WWPP H.11
2 one = ['for', 'return', 'pass', 'def', 'bool']
3 two = one
4 three = one[:]
5
6 two[0] = 'return'
7 three[3] = 'return'
8
9 count = 0
10 for key in [one, two, three]:
11     if key[0] == 'return':
12         count += 1
13     if key[1] == 'return':
14         count += 1
15
16 print (count)
```

Show hidden output

```
1 #@title
2 #WWPP H.12
3 nums = [1, 2, 3, 4, 5]
4 #nums = [2, 4, 6]
5
6 y = [x+1 for x in nums if x%2 != 0]
7 print(sum(y))
```

```
1 #WWPP H.13
2 num1 = [1, 2, 3]
3 num2 = [1, 3, 2]
4 print(1) if (num1 == num2) else print(2)
```

Show hidden output

```
1 #WWPP H.14
2 nums = [88, 2, 3, 99]
3 a, b, c, d = nums
4 x = max([a,c])
5 print(x)
```

Show hidden output

```
#WWPP H.15
nums = [88, 2, 3, 99]
a, *b, c = nums
x = sum(b)
print(x)
```

Show hidden output

```
#WWPP H.16
nums = [1,2,3]
vals = nums
vals.append(4)
print(len(nums))
```

Show hidden output

```
#WWPP H.17
nums = [1,2,3]
vals = nums[:]
vals.append(4)
print(sum(nums))
```

Show hidden output

```
#WWPP H.18
nums = [1,2,3, 77]
vals = nums.copy()
vals = vals + [88, 99]
print(max(nums))
```

Show hidden output

```
#WWPP H.19
chars = ['a', 'b', 'c', 'a', 'd']
chars[1] = 'r'
chars[2] = chars[1]
if ('c' in chars):
  x = chars[4]
else:
  x = chars[2]

print(x)
```

Show hidden output

```
#WWPP H.20
a = list(range(1, 11))
a[1:4]
```

```
x = len(a)
print(x)
```

Show hidden output

```
#WWPP H.21
a = ['cat', 'ox', 'snake']
b = max(a)
c = min(a)
x = c[1] + b[2]
print(x)
```

Show hidden output

```
#WWPP H.22
chars = ['a', 'x', 'y', 'y', 'b', 'p']
x = chars[1:3] + ['x', 'z']
print(x.count('x'))
```

Show hidden output

```
#WWPP H.23
num1 = [1, 2, 3, 4, 5, 6]
num2 = num1[-1:-5:-2]
print(min(num2))
5
```

Show hidden output

```
#WWPP H.24
nums = [1, 2, -1, -2, 3]
x = nums[nums[-1]]
x = nums[x]
print(x)
```

Show hidden output

```
#WWPP H.25
num1 = [ [1,2,3], [4,5,6], [7,8,9], [1,2,3]]
x = num1[0][1]
y = num1[1][0]
print(x+y)
```

Show hidden output

```
#WWPP H.26
my_list = [1, 2]

for v in range(2):
  my_list.insert(v, my_list[v])
```

```
6
7 print(my_list)
```

Show hidden output

```
1 #WWPP H.27
2 nums = [0,1]
3 for i in range(len(nums)):
4   nums.insert(0, i)
5
6 print(*nums)
7
```

Show hidden output

```
1 #WWPP H.28 (nested list)
2 my_list = [8, [3.41, 20, [30, 'foo', True]], ['bar', 'xyz']]
3 x = my_list[-1][0][1:3]
4 print(x)
```

Show hidden output

```
1 #WWPP H.29 (nested list)
2 my_list = [8, [3.41, 20, [30, 'foo', True]], 'bar']
3 x = my_list[1][2][1]
4 print(x)
```

Show hidden output

```
1 #WWPP H.30 (list and functions)
2 def fun_one(x):
3     x += [10]
4     return x
5
6 x = fun_one([2,3])
7 print (len(x))
8
```

Show hidden output

```
1 #WWPP H.31 (list and functions)
2
3 a = ['python', 'programming', 'is', 'cool']
4
5 def foo(x):
6   del x[2]
7   x[2] = 'rocks'
8
9 foo(a)
```

```
print(a)
```

Show hidden output

I. Tuples

```
#WWPP I.1
a = (3)
b = (4,)
c = (5,6)

x = 0
if type(a) is tuple:
  x += 1
if type(b) is tuple:
  x += 1
if type(c) is tuple:
  x += 1

print(x)
```

Show hidden output

```
#WWPP I.2
a = (1, 3, 99, 45, 76)
b = a[-2:]
b += (99, )
x = len(b)
print(x)
```

Show hidden output

```
#WWPP I.3
a = (4, 3, 2)
x = a + (1,2)
print(x.count(2))
```

Show hidden output

```
#WWPP I.4
tuple5 = ((1, 2), [3, 4], (5, 6))
tuple5[1][0] = 9
x = sum(tuple5[1])
print(x)
```

Show hidden output

```
#WWPP I.5
nums = (10, 20, 30, 40, 50)
x = nums[2:4]
y = nums[:3]
z = nums[3:]

total = len(x) + len(y) + len(z)
print(total)
```

Show hidden output

J. Sets

```
#WWPP J.1
set1 = {10, 20, 30, 40, 50}
set2 = {60, 70, 10, 30, 40, 80, 20, 50}

x = set1.issubset(set2)
y = set2.issuperset(set1)

print(x and y)
```

```
#WWPP J.2
my_set = {1, 0, True, False, 1, 2}
print(len(my_set))
```

Show hidden output

```
#WWPP J.3
set1 = {1, 2, 3}
set2 = {2, 3, 3, 4}

set3 = set1.union(set2)
print(sum(set3))
```

```
#WWPP J.4
x = 10

set1 = {1, 2, 3, 9, 5, 6, 7, 8, 10}
set2 = {2, 3, 3, 4}

a = set1.union(set2)
b = set2.union(set1)
if (a==b):
  x += 10
```

```
11
12 print(x)
```

```
 1 #WWPP J.5
 2 x = 10
 3
 4 set1 = {1, 2, 3, 9, 5, 6, 7, 8, 10}
 5 set2 = {2, 3, 3, 4}
 6
 7 a = set1.difference(set2)
 8 b = set2.difference(set1)
 9 if (a==b):
10   x += 10
11
12 print(x)
```

```
 1 #WWPP J.6
 2
 3 a = [4, 5, 5, 6]
 4 b = [6, 4, 4, 5]
 5 a = set(a)
 6 b = set(b)
 7
 8 if (a==b):
 9   print(10)
10 else:
11   print(20)
```

```
 1 #WWPP J.7
 2 x = {4,5,6}
 3 y = {5,6,7}
 4
 5 print( (x&y) < y)
```

K. Dictionaries

```
 1 #WWPP K.1
 2 a = {}
 3 a['1'] = 300
 4 a["1"] = 100
 5 a[1] = 200
 6 print(a['1'])
 7
```

```
#WWPP K.2
a = {1:2, 2:20, 2:21}

for x, y in a.items():
    pass

total = 0
for value in a.values():
    total += value

print(x + y)
print(total)

```

Show hidden output

```
#WWPP K.3
school = {
          "class":
                  {
                  "student":
                      {
                        "name":"Mike",
                        "marks":{
                                  "quizzes":[10, 20, 9, 5],
                                  "tests":[80, 89, 92]
                                }
                      }
                   }
         }

x = school['class']['student']['marks']['quizzes'][1]
print(x)
```

```
#WWPP K.4
x = [1, 2, {1:2, 2:3}, 3]
y = x[2][2]
print(y)
```

```
#WWPP K.5
a = [
      'a',
      'b',
      {
          'foo': 1,
```

```
          'bar':
                {
                    'x' : 10,
                    'y' : 20,
                    'z' : 30
                },
          'baz': 3
      },
      'c',
      'd'
   ]

x = a[2]['bar']['z']
print(x)
```

Show hidden output

```
#WWPP K.6
a = {1:2, 2:-1, 3:1, 4:1}
b = sum([x for x in a.values()])
print(a[b])
```

Show hidden output

```
#WWPP K.7
a = {'a':'b', 'b':'c', 'c':'a'}
x = a[a['a']]
print(x)
```

```
#WWPP K.8
a = {1:1, 2:2, 3:3}

list_1 = list(a.values())
list_2 = list(a.keys())

print(10) if (list_1 == list_2) else print(20)
```

```
#WWPP K.9
a = {1:1, 2:2, 3:3}
x = (a.values() == a.keys())
print(10) if x else print(20)
```

```
#WWPP K.10
dct = {}
dct['1'] = (2, 3)
dct['4'] = (5, 6)
```

```
5
for x in dct.keys():
   print(dct[x][1], end = '')
```

Show hidden output

L. Bitwise Operators

```
#WWPP L.1
#hex #oct #scientific notation
a, b, c, d = 0, 1, 0, 1
p =  (a & b)
q =  (b | c)
r =  (b ^ d)

print(p, q, r,  sep = '')
```

```
010
```

```
#WWPP L.2
a = 2
x = a << 1
y = a << 2
print(x+y)
```

```
12
```

```
#WWPP L.3
a = 32
x = a >> 1
y = x >> 1
print(x+y)
```

Show hidden output

```
#WWPP L.4
a = '0b1111'
x = int(a,2)
print(x)
```

Show hidden output