tuple1 = (1120, 'a')

**print**(**max**(tuple1))

TypeError  
1120  
‘a’

aTuple = ("Orange", [10, 20, 30], (5, 15, 25))

Choose the correct way to access value **20** from the following tüple  
aTuple[1:2][1]  
aTuple[1:2](1)  
aTuple[1:2][1]  
aTuple[1][1]

Select which is true for Python tüple  
A tuple maintains the order of items  
 A tuple is unordered  
We cannot change the tuple once created  
We can change the tuple once created

my\_list = ["Hello", "Python"]

**print**("-".join(my\_list))

HelloPython-  
Hello-Python  
-HelloPython

In Python, list is mutable

 False

 True

Dictionary keys must be immutable

 True

 False

Select the correct way to access the value of a history subject

sampleDict = {

"class":{

"student":{

"name":"Mike",

"marks":{

"physics":70,

"history":80

}

}

}

}

sampleDict['class']['student']['marks']['history']  
sampleDict['class']['student']['marks'][1]  
sampleDict['class'][0]['marks']['history']

dict1 = {"key1":1, "key2":2}

dict2 = {"key2":2, "key1":1}

**print**(dict1 == dict2)

 True

 False

9. In Python, Dictionaries are immutable

 False

 True

11. Select correct ways to create an empty dictionary

 sampleDict = {}

 sampleDict = dict()

 sampleDict = dict{}

**import** math

**print**(math.ceil(252.4))

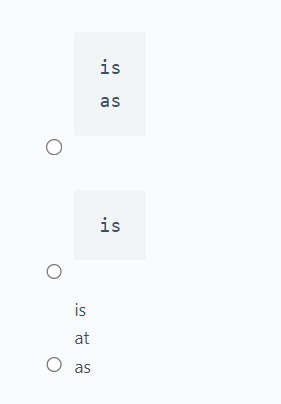
**print**(math.floor(252.4))

 252  
252

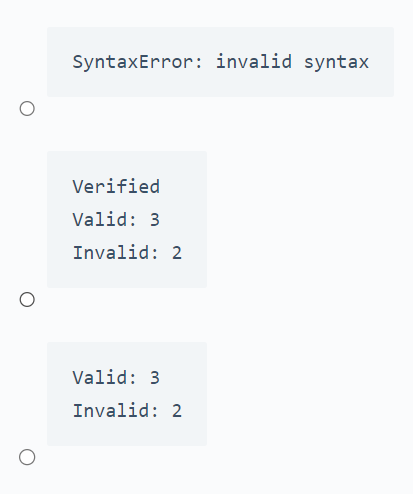
252  
253

 253  
252

words = ["is", "was", "this", "that", "at"]  
for item in words:  
    if(len(item) > 2):  
        continue  
    else:  
        print(item)  
        break  
else:  
    print("as")



items = [1, -1, 3, 4, -5]  
  
valid = 0  
invalid = 0  
  
for i in items:  
    if(i > 0):  
        valid = valid + 1  
    else:  
        invalid = invalid + 1  
else:  
    print("Verified")  
  
print("Valid: " + str(valid))  
print("Invalid: " + str(invalid))



def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n - 1)

result = factorial(5)

print(result)

# Soru: Aşağıdaki kod bloğu çalıştırıldığında çıktısı ne olur?

def find\_duplicates(lst):

seen = set()

duplicates = set()

for item in lst:

if item in seen:

duplicates.add(item)

else:

seen.add(item)

return list(duplicates)

my\_list = [1, 2, 3, 4, 2, 5, 6, 3]

result = find\_duplicates(my\_list)

print(result)

def count\_vowels(text):

vowels = "aeiouAEIOU"

count = 0

for char in text:

if char in vowels:

count += 1

return count

result = count\_vowels("Hello World")

print(result)

def is\_palindrome(s):

return s == s[::-1]

text = "radar"

result = is\_palindrome(text)

print(result)

def merge\_dicts(\*dicts):

result = {}

for d in dicts:

result.update(d)

return result

dict1 = {'a': 1, 'b': 2}

dict2 = {'b': 3, 'c': 4}

dict3 = {'d': 5}

result = merge\_dicts(dict1, dict2, dict3)

print(result)