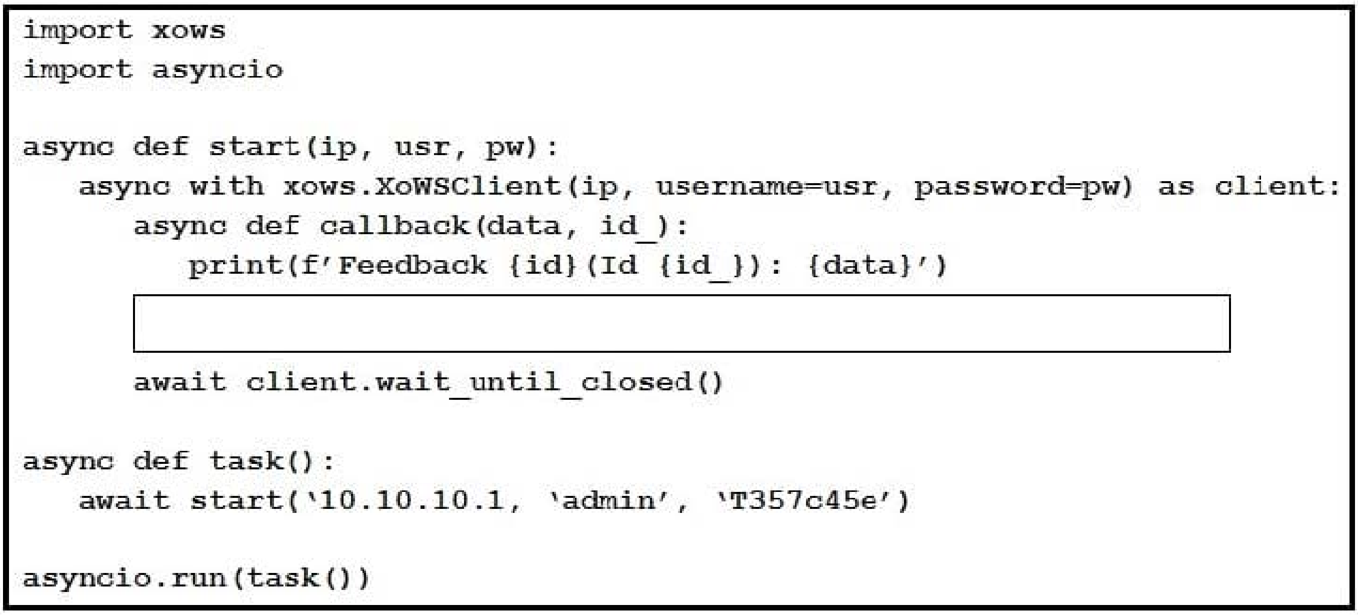
300-835 Q&As

Automating Cisco Collaboration Solutions (CLAUTO)

Pass Cisco 300-835 Exam with 100% Guarantee

QUESTION 1



```
import xows
import asyncio

async def start(ip, usr, pw):
   async with xows.XoWSClient(ip, username=usr, password=pw) as client:
      async def callback(data, id_):
         print(f'Feedback {id}(Id {id_}): {data}')


      await client.wait_until_closed()

async def task():
   await start('10.10.10.1, 'admin', 'T357c45e')

asyncio.run(task())
```

Refer to the exhibit. This Python script uses the websocket-based, xAPI library "pyxows\\' to monitor and print event details to the console when users interact with UI Extension/In-Room Control `Action Buttons\\' deployed to the Touch-10 interface of a Webex room device. Which two code snippets successfully capture such events? (Choose two.)

A. await client.xEvent([`Event\\', `**\\'], callback)

B. await client.subscribe([`Event\\', `ActionButton\\', `Clicked\\'], callback)

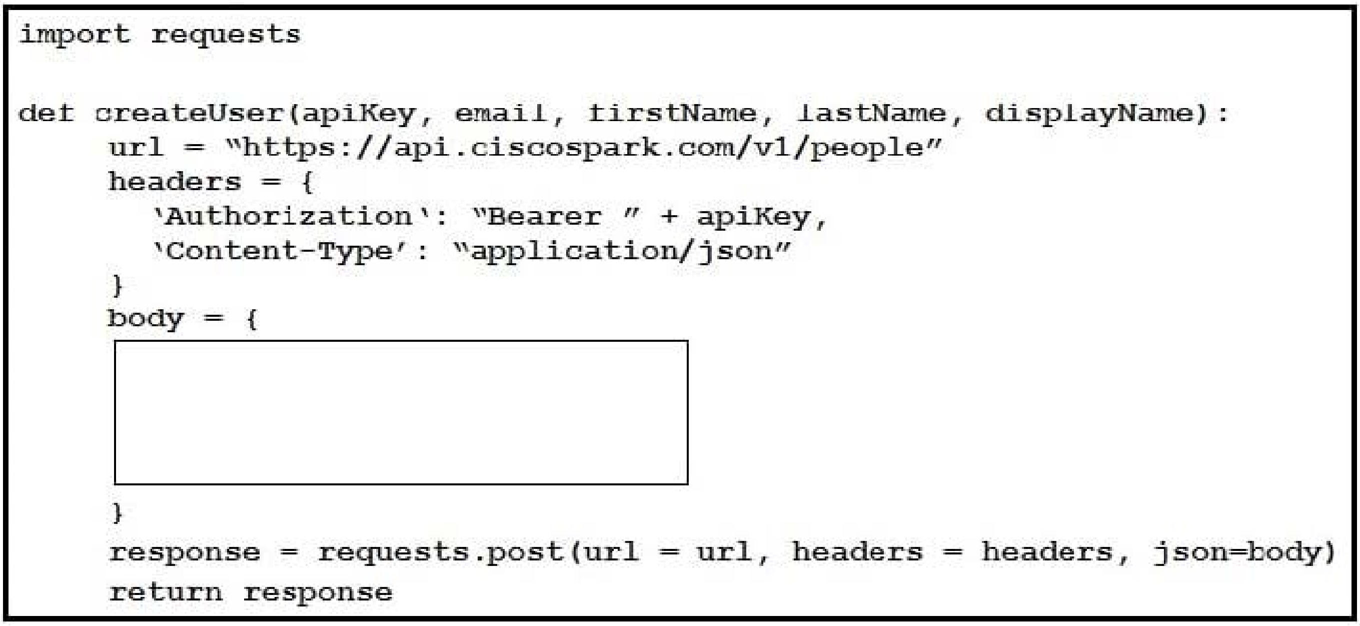
C. await client.subscribe([`Event\\', `**\\'], callback)

D. await client.xEvent([`Event\\', `UserInterface\\', `Extensions\\', `Panel\\', `Clicked\\'], callback)

E. await client.subscribe([`Event\\', `UserInterface\\', `Extensions\\', `Panel\\', `Clicked\\'], callback)

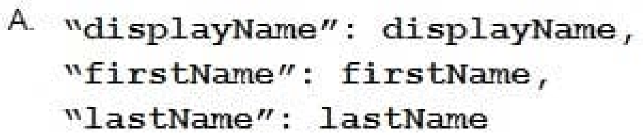
Correct Answer: DE

QUESTION 2

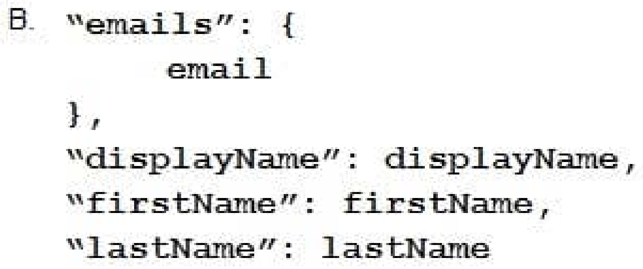


```
import requests

def createUser(apiKey, email, firstName, lastName, displayName):
     url = “https://api.ciscospark.com/v1/people”
     headers = {
        ‘Authorization‘: “Bearer ” + apiKey,
        ‘Content-Type’: “application/json”
     }
     body = {



     }
     response = requests.post(url = url, headers = headers, json=body)
     return response
```

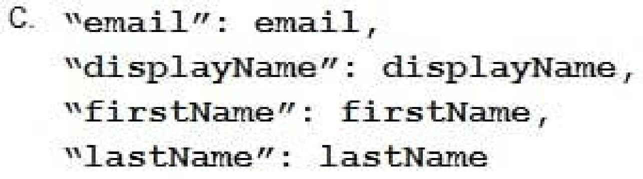
Refer to the exhibit. Which snippet of code does a user with the administrator role use in the missing "body" section to create a new user in a Webex Teams organization?



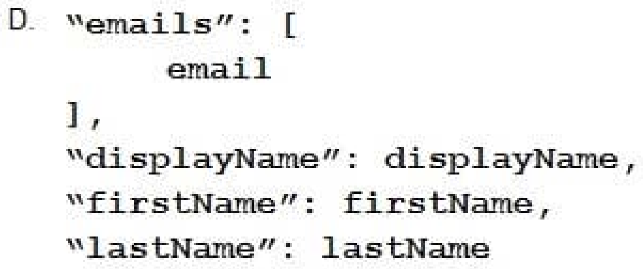
```
A. “displayName”: displayName,
   “firstName”: firstName,
   “lastName”: lastName
```



```
B. “emails”: {
        email
   },
   “displayName”: displayName,
   “firstName”: firstName,
   “lastName”: lastName
```



```
C. “email”: email,
   “displayName”: displayName,
   “firstName”: firstName,
   “lastName”: lastName
```



```
D. “emails”: [
        email
   ],
   “displayName”: displayName,
   “firstName”: firstName,
   “lastName”: lastName
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

QUESTION 3

Which statement describe the function of the perfmonCollectCounterData operation used in the Cisco Unified Communications Manager Serviceability Perfmon API?

A. It returns the Perfmon data for all counters that belong to an object on a particular host.

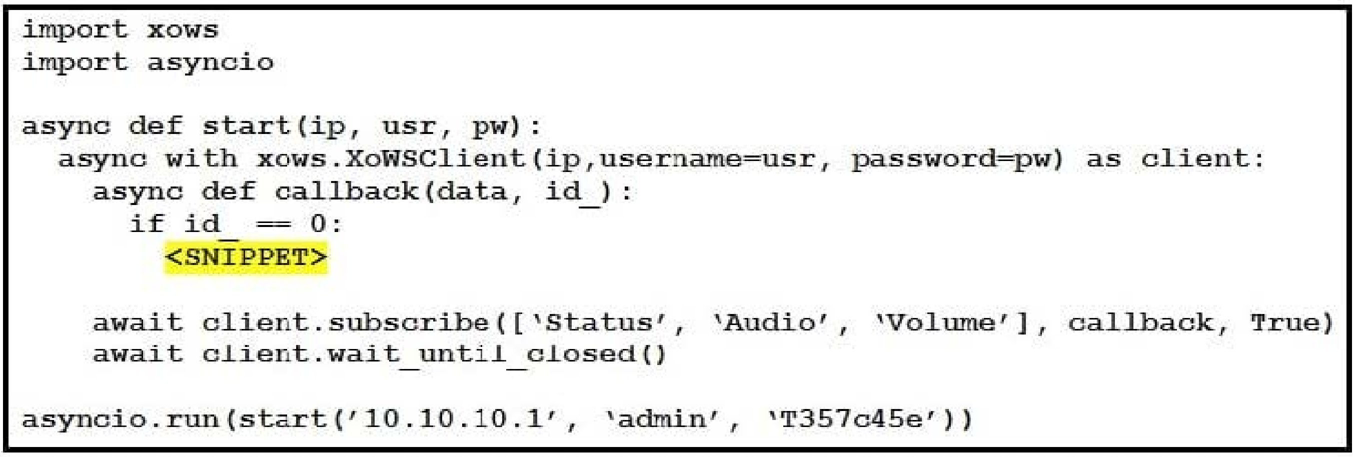
B. It collects files from RTMT.

C. It collects the Perfmon data for all counters that have been added with perfmonAddCounter to the session handle returned from perfmonOpenSession.

D. It returns the list of available Perfmon objects and counters on a particular host.

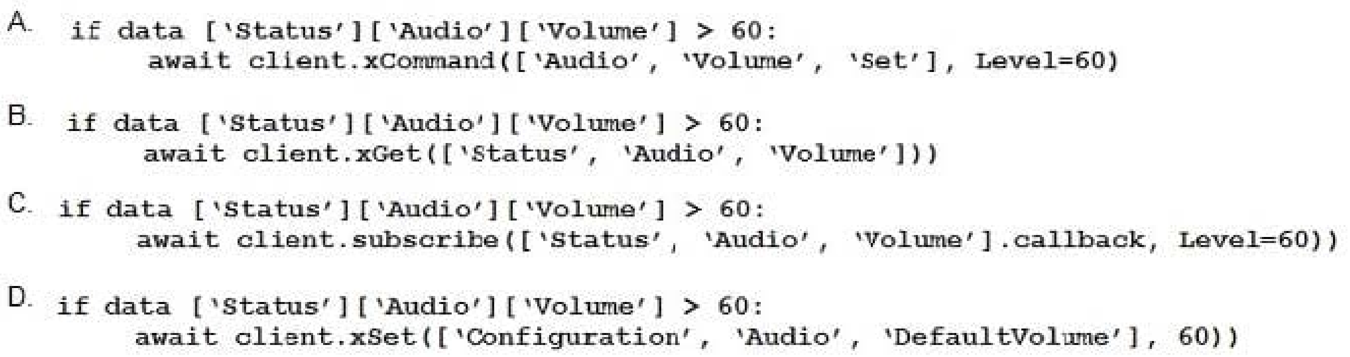
Correct Answer: A

QUESTION 4



```
import xows
import asyncio

async def start(ip, usr, pw):
  async with xows.XoWSClient(ip,username=usr, password=pw) as client:
    async def callback(data, id_):
      if id_ == 0:
        <SNIPPET>

    await client.subscribe([‘Status’, ‘Audio’, ‘Volume’], callback, True)
    await client.wait_until_closed()

asyncio.run(start(’10.10.10.1’, ‘admin’, ‘T357c45e’))
```

Refer to the exhibit. An administrator is creating a script using the Python xAPI over WebSockets(pyxows) library. The goal of this script is to monitor the volume of the endpoint and set the volume to 60 whenever the volume has been set higher than that amount. Which code snippet accomplishes this task when it is added?



```
A. if data [‘Status’][‘Audio’][‘Volume’] > 60:
        await client.xCommand([‘Audio’, ‘Volume’, ‘Set’], Level=60)

B. if data [‘Status’][‘Audio’][‘Volume’] > 60:
        await client.xGet([‘Status’, ‘Audio’, ‘Volume’]))

C. if data [‘Status’][‘Audio’][‘Volume’] > 60:
        await client.subscribe([‘Status’, ‘Audio’, ‘Volume’].callback, Level=60))

D. if data [‘Status’][‘Audio’][‘Volume’] > 60:
        await client.xSet([‘Configuration’, ‘Audio’, ‘DefaultVolume’], 60))
```

A. Option A

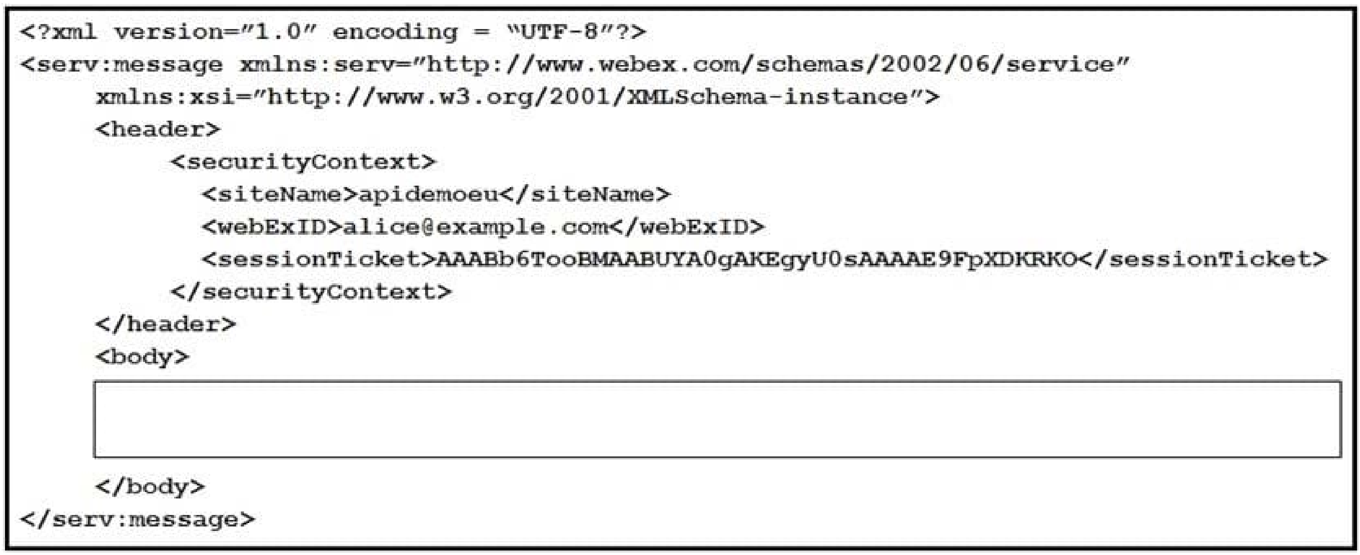
B. Option B

C. Option C

D. Option D

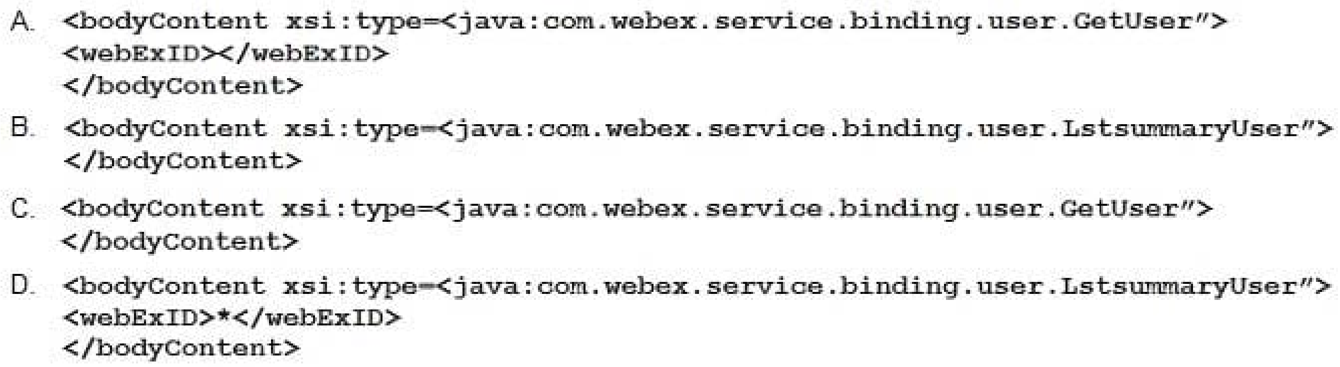
Correct Answer: A

QUESTION 5



```
<?xml version=”1.0” encoding = “UTF-8”?>
<serv:message xmlns:serv=”http://www.webex.com/schemas/2002/06/service”
     xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”>
     <header>
          <securityContext>
            <siteName>apidemoeu</siteName>
            <webExID>alice@example.com</webExID>
            <sessionTicket>AAABb6TooBMAABUYA0gAKEgyU0sAAAAE9FpXDKRKO</sessionTicket>
          </securityContext>
     </header>
     <body>

     </body>
</serv:message>
```

Refer to the exhibit. Alice is a Webex Meetings admin for her organization. Which code snippet completes a Webex Meetings XML API request that returns all users?



A.
```
<bodyContent xsi:type=<java:com.webex.service.binding.user.GetUser”>
<webExID></webExID>
</bodyContent>
```

B.
```
<bodyContent xsi:type=<java:com.webex.service.binding.user.LstsummaryUser”>
</bodyContent>
```

C.
```
<bodyContent xsi:type=<java:com.webex.service.binding.user.GetUser”>
</bodyContent>
```

D.
```
<bodyContent xsi:type=<java:com.webex.service.binding.user.LstsummaryUser”>
<webExID>*</webExID>
</bodyContent>
```

A. Option A

B. Option B

C. Option C

D. Option D

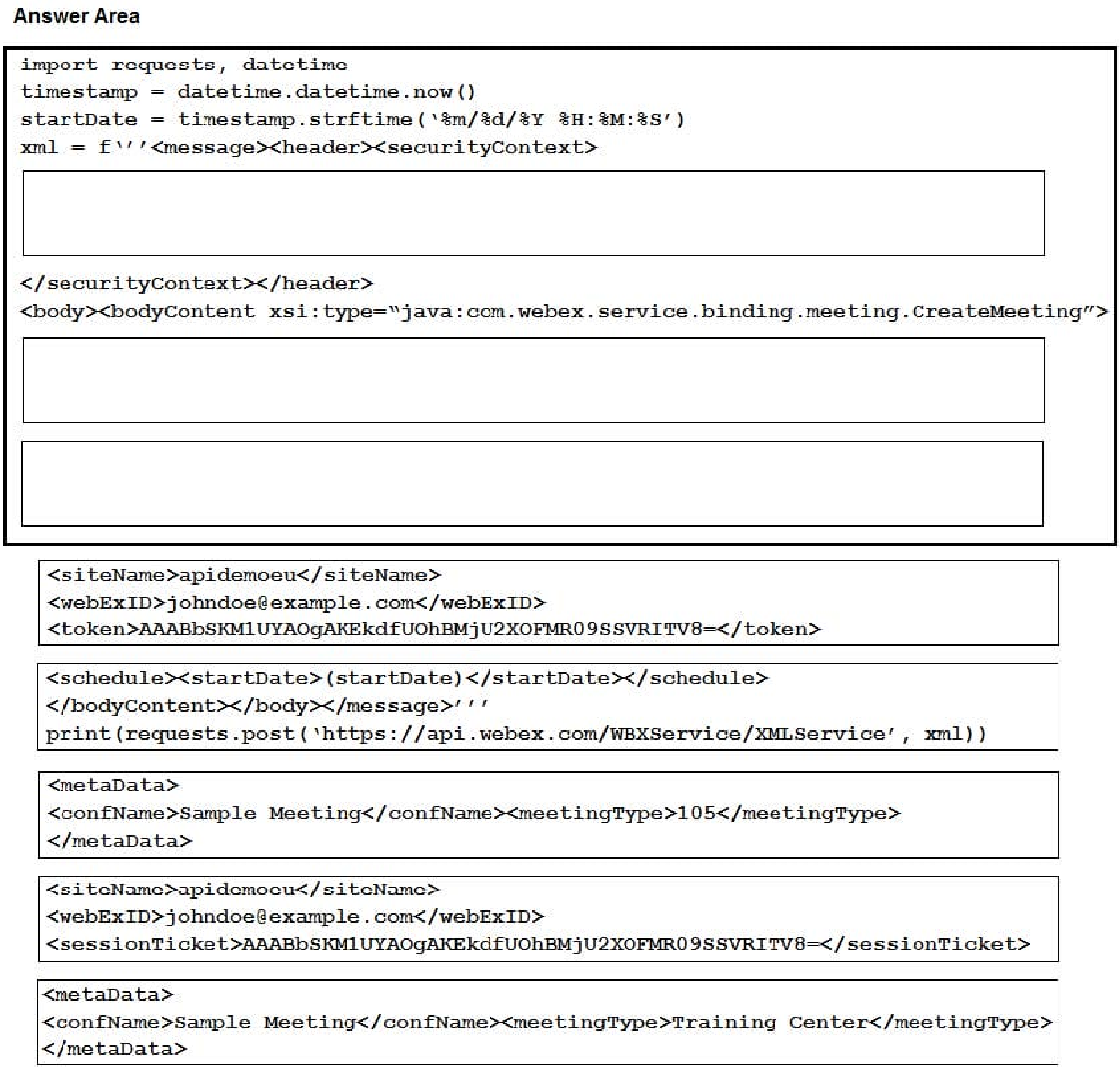
Correct Answer: A

QUESTION 6

DRAG DROP

A small Python script is constructed that creates a Webex Meeting for John Doe scheduled for the current time. Drag and drop the code snippets at the bottom onto the areas of the source code exhibit to create a program. Not all options are used.

Select and Place:



Answer Area

```
import requests, datetime
timestamp = datetime.datetime.now()
startDate = timestamp.strftime('%m/%d/%Y %H:%M:%S')
xml = f'''<message><header><securityContext>
```

```
</securityContext></header>
<body><bodyContent xsi:type="java:com.webex.service.binding.meeting.CreateMeeting">
```

```
<siteName>apidemoeu</siteName>
<webExID>johndoe@example.com</webExID>
<token>AAABbSKM1UYAOgAKEkdfUOhBMjU2XOFMR09SSVRITV8=</token>
```

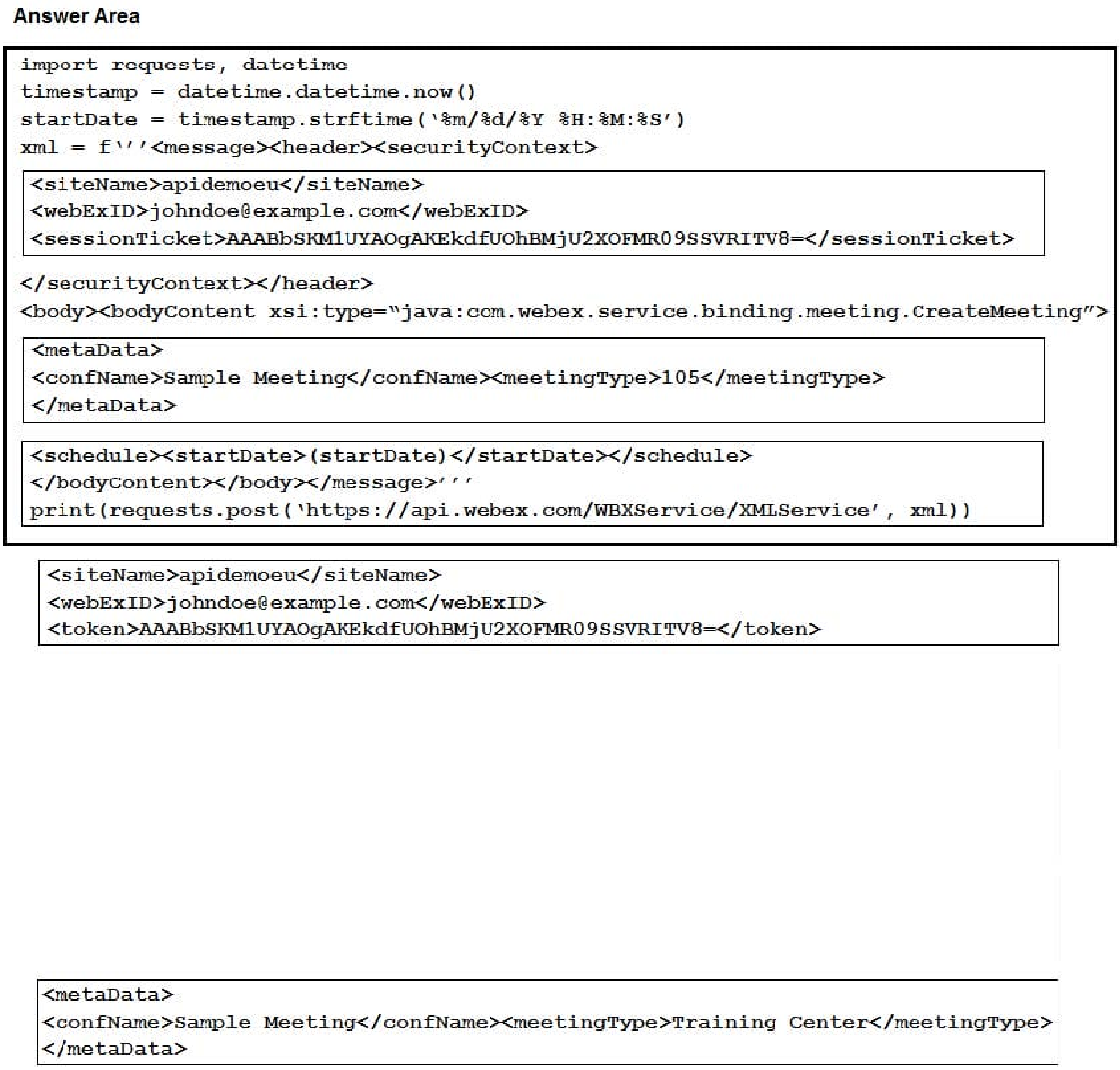
```
<schedule><startDate>(startDate)</startDate></schedule>
</bodyContent></body></message>'''
print(requests.post('https://api.webex.com/WBXService/XMLService', xml))
```

```
<metaData>
<confName>Sample Meeting</confName><meetingType>105</meetingType>
</metaData>
```

```
<siteName>apidemoeu</siteName>
<webExID>johndoe@example.com</webExID>
<sessionTicket>AAABbSKM1UYAOgAKEkdfUOhBMjU2XOFMR09SSVRITV8=</sessionTicket>
```

```
<metaData>
<confName>Sample Meeting</confName><meetingType>Training Center</meetingType>
</metaData>
```

Correct Answer:



Answer Area

```
import requests, datetime
timestamp = datetime.datetime.now()
startDate = timestamp.strftime('%m/%d/%Y %H:%M:%S')
xml = f'''<message><header><securityContext>
```

```
<siteName>apidemoeu</siteName>
<webExID>johndoe@example.com</webExID>
<sessionTicket>AAABbSKM1UYAOgAKEkdfUOhBMjU2XOFMR09SSVRITV8=</sessionTicket>
```

```
</securityContext></header>
<body><bodyContent xsi:type="java:com.webex.service.binding.meeting.CreateMeeting">
```

```
<metaData>
<confName>Sample Meeting</confName><meetingType>105</meetingType>
</metaData>
```

```
<schedule><startDate>(startDate)</startDate></schedule>
</bodyContent></body></message>'''
print(requests.post('https://api.webex.com/WBXService/XMLService', xml))
```

```
<siteName>apidemoeu</siteName>
<webExID>johndoe@example.com</webExID>
<token>AAABbSKM1UYAOgAKEkdfUOhBMjU2XOFMR09SSVRITV8=</token>
```

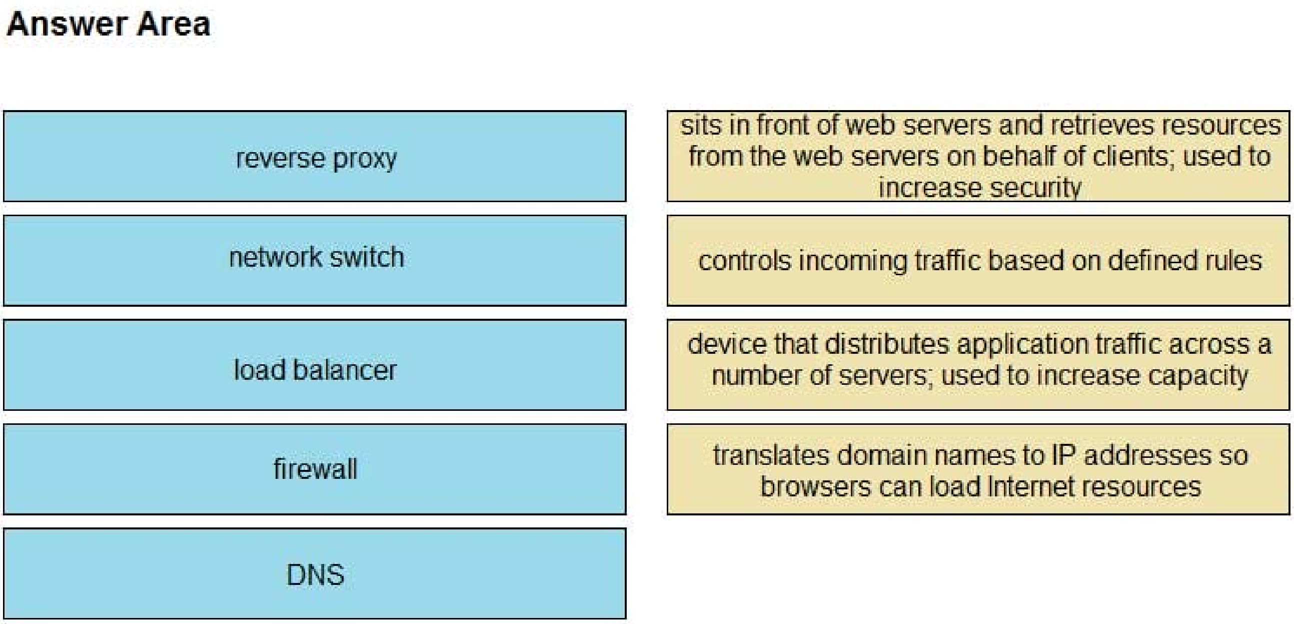
```
<metaData>
<confName>Sample Meeting</confName><meetingType>Training Center</meetingType>
</metaData>
```

QUESTION 7

DRAG DROP

Drag and drop the elements from the left onto the correct descriptions on the right. Not all options are used.

Select and Place:



Answer Area

reverse proxy

network switch

load balancer

firewall

DNS

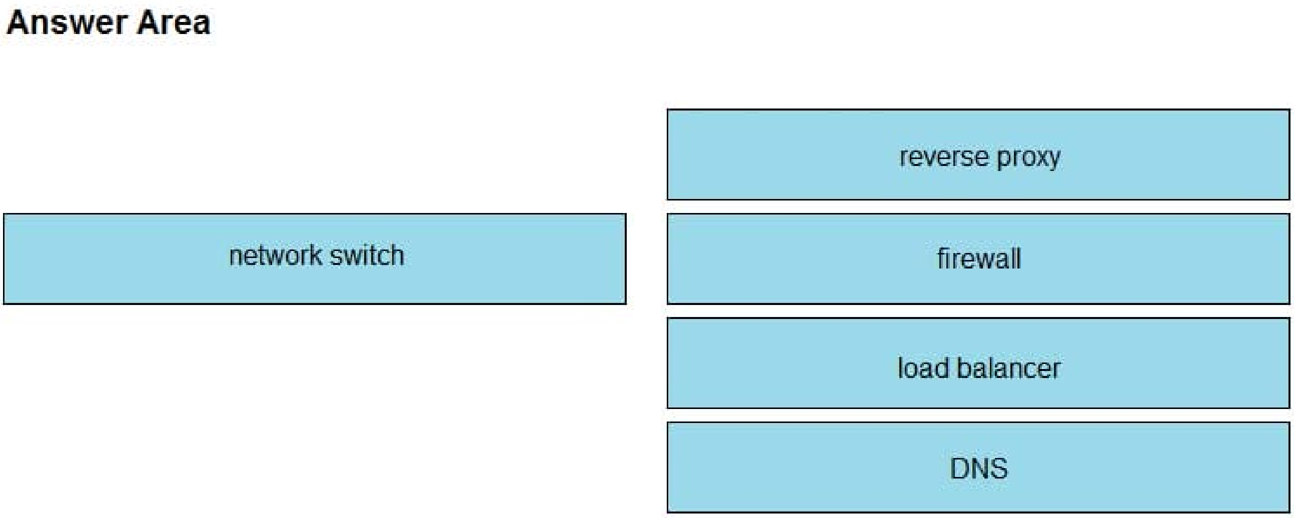
sits in front of web servers and retrieves resources from the web servers on behalf of clients; used to increase security

controls incoming traffic based on defined rules

device that distributes application traffic across a number of servers; used to increase capacity

translates domain names to IP addresses so browsers can load Internet resources

Correct Answer:



Answer Area

network switch

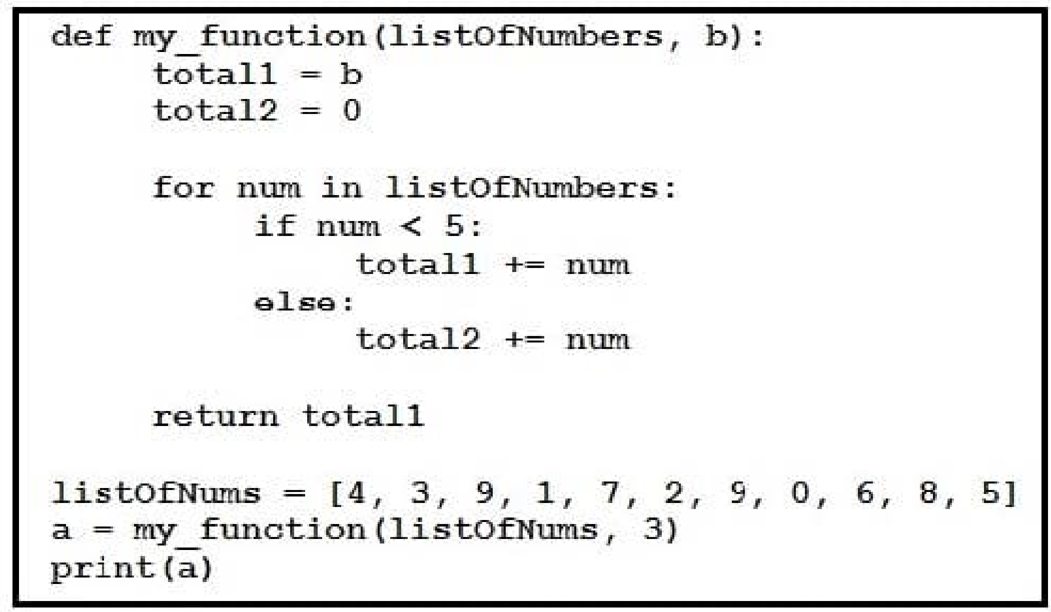
reverse proxy

firewall

load balancer

DNS

QUESTION 8



```
def my_function(listOfNumbers, b):
    total1 = b
    total2 = 0

    for num in listOfNumbers:
        if num < 5:
            total1 += num
        else:
            total2 += num

    return total1

listOfNums = [4, 3, 9, 1, 7, 2, 9, 0, 6, 8, 5]
a = my_function(listOfNums, 3)
print(a)
```

Refer to the exhibit. Which value of "a" is displayed when the Python code is executed?

A. 10

B. 13

C. 18

D. 21

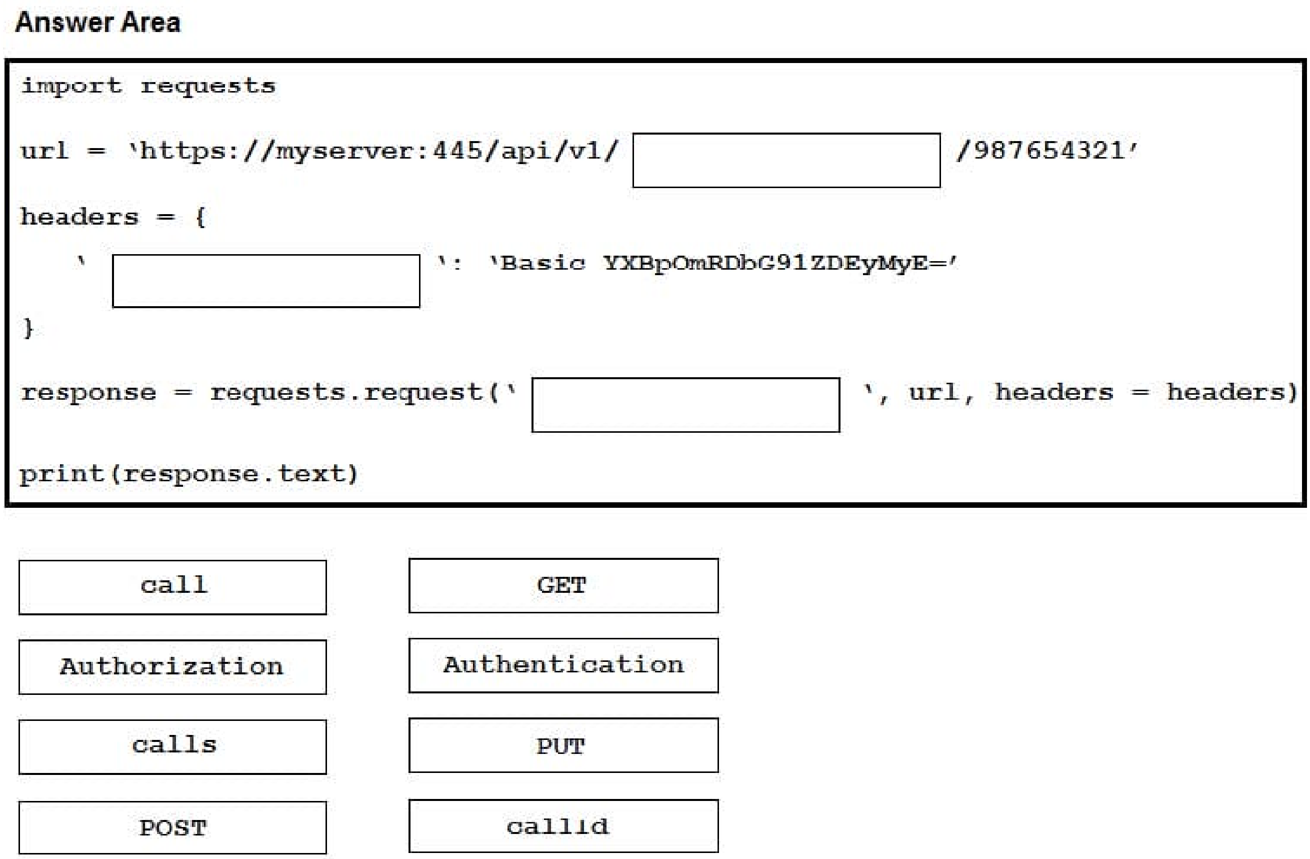
Correct Answer: B

QUESTION 9

DRAG DROP

The Python script executes a REST API to retrieve information from the Cisco Meeting Server about the call with an identifier of “987654321”. Drag and drop the code onto the snippet to complete the Python script. Not all options are used.

Select and Place:

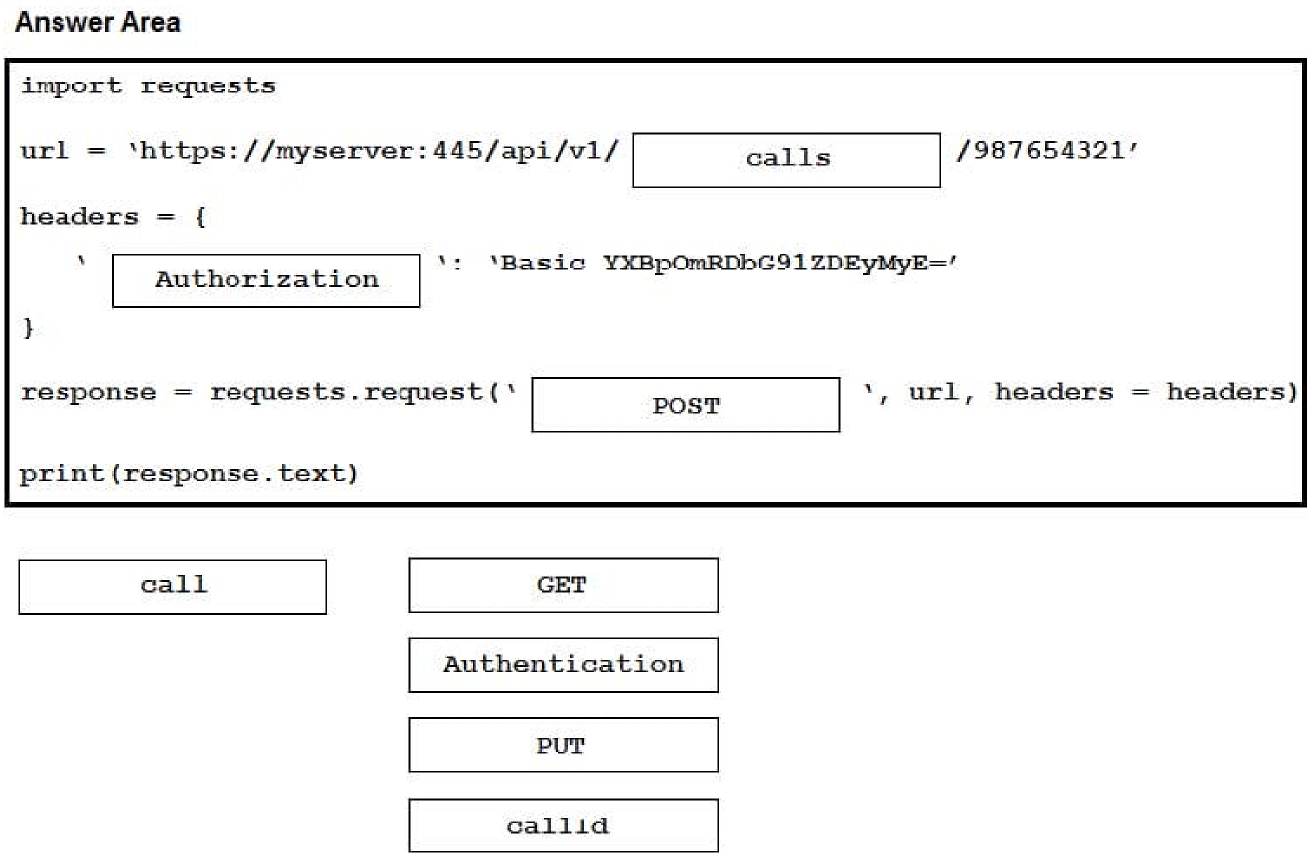


Answer Area

```
import requests

url = 'https://myserver:445/api/v1/                /987654321'

headers = {
    '                ': 'Basic YXBpOmRDbG91ZDEyMyE='
}

response = requests.request('                ', url, headers = headers)

print(response.text)
```

call	GET
Authorization	Authentication
calls	PUT
POST	callid

Correct Answer:



Answer Area

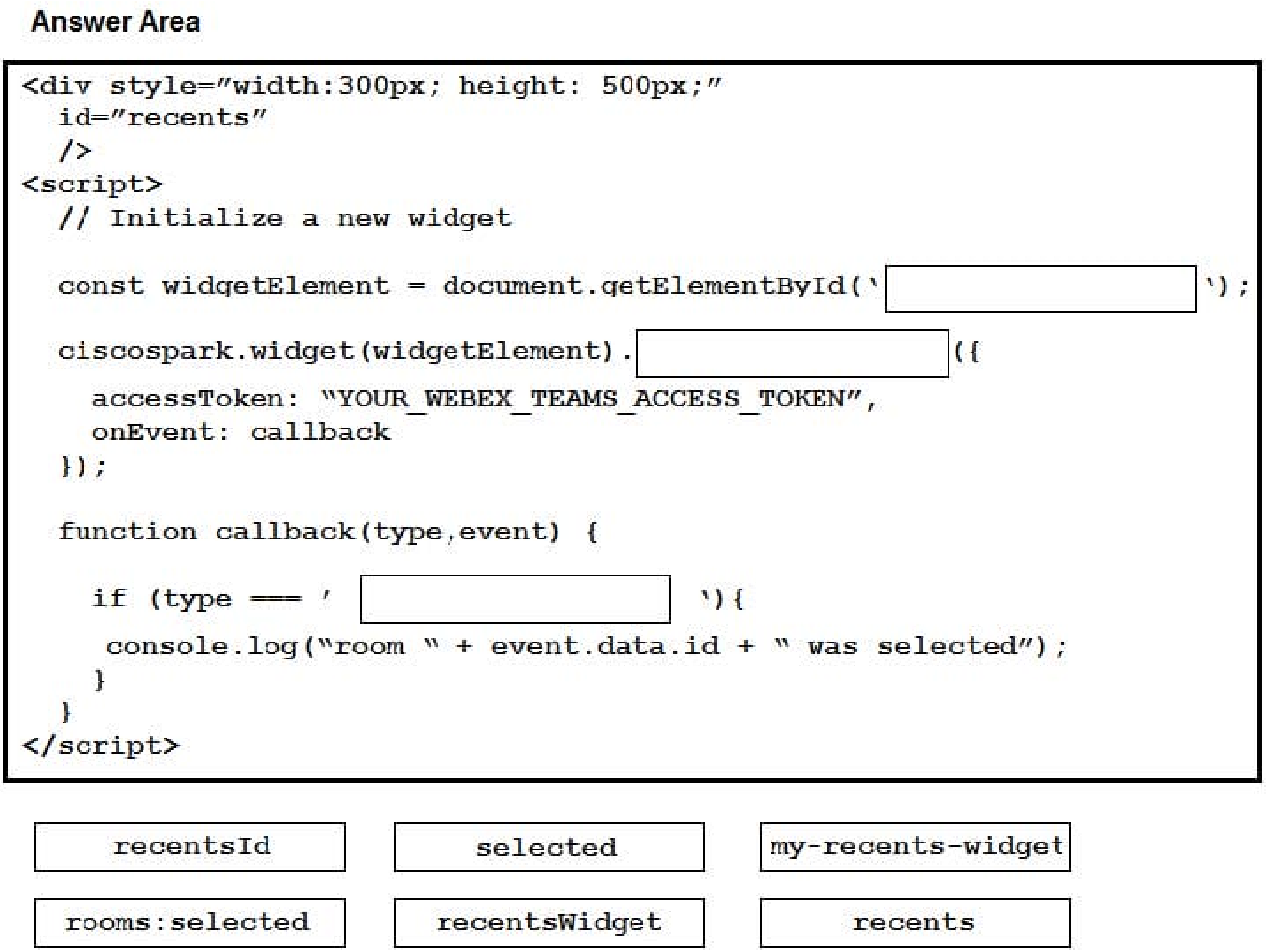
```
import requests

url = 'https://myserver:445/api/v1/ calls /987654321'

headers = {
    ' Authorization ': 'Basic YXBpOmRDbG91ZDEyMyE='
}

response = requests.request(' POST ', url, headers = headers)

print(response.text)
```

QUESTION 10

DRAG DROP

Drag and drop the code to create a Webex Teams Recents Widget in an existing HTML page. Not all items are used.

Select and Place:

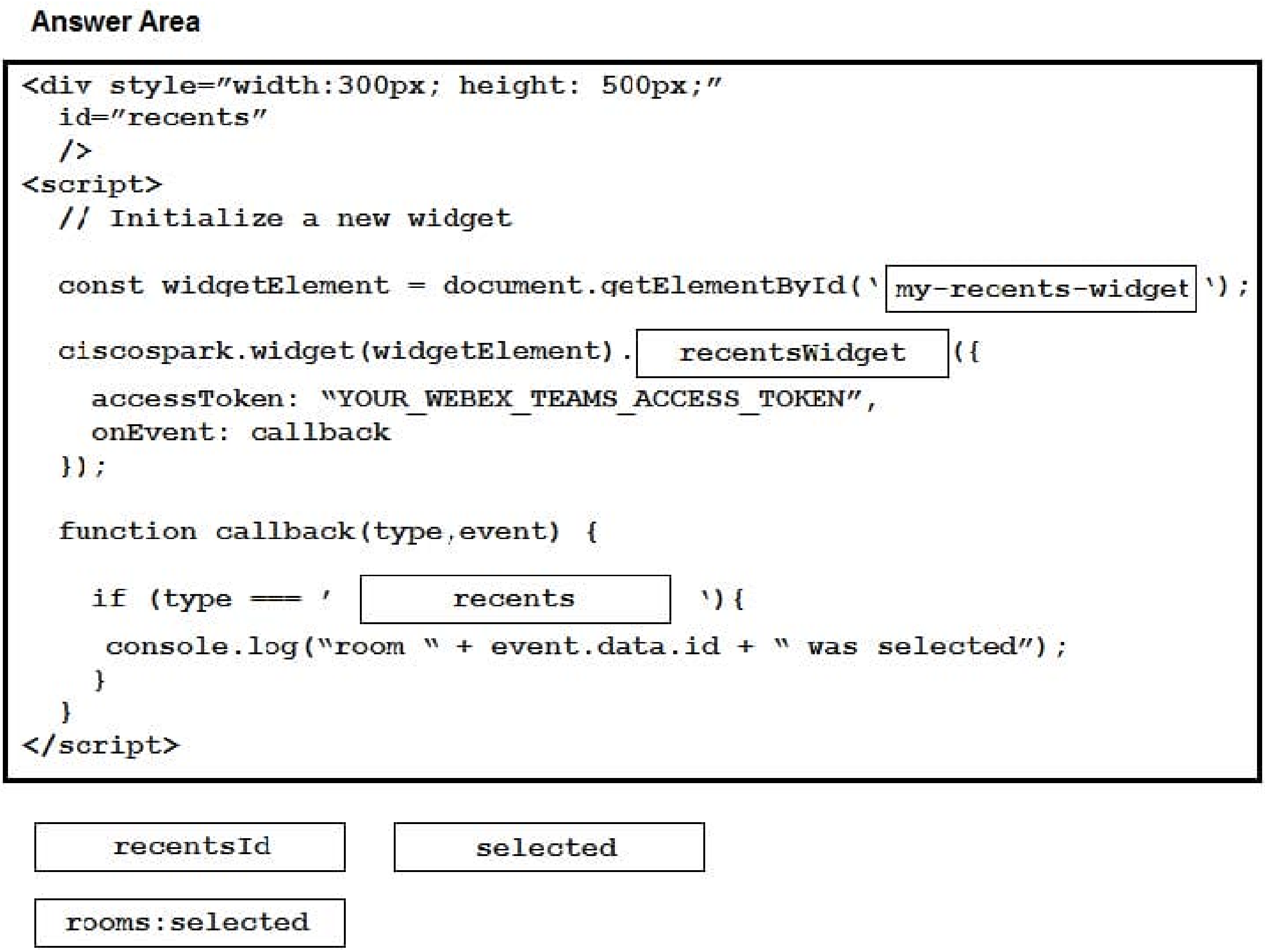


Answer Area

```
<div style=”width:300px; height: 500px;”
  id=”recents”
  />
<script>
  // Initialize a new widget

  const widgetElement = document.getElementById(‘□’);

  ciscospark.widget(widgetElement).□({
    accessToken: “YOUR_WEBEX_TEAMS_ACCESS_TOKEN”,
    onEvent: callback
  });

  function callback(type,event) {

    if (type === ’□‘){
     console.log(“room ” + event.data.id + ” was selected”);
    }
  }
</script>
```

recentsId	selected	my-recents-widget
rooms:selected	recentsWidget	recents

Correct Answer:



Answer Area

```
<div style="width:300px; height: 500px;"
  id="recents"
  />
<script>
  // Initialize a new widget

  const widgetElement = document.getElementById('my-recents-widget');

  ciscospark.widget(widgetElement).recentsWidget({
    accessToken: "YOUR_WEBEX_TEAMS_ACCESS_TOKEN",
    onEvent: callback
  });

  function callback(type,event) {

    if (type === 'recents'){
     console.log("room " + event.data.id + " was selected");
    }
  }
</script>
```

recentsId

selected

rooms:selected

[Latest 300-835 Dumps](https://www.leads4pass.com/300-835.html) [300-835 Practice Test](https://www.leads4pass.com/300-835.html) [300-835 Exam Questions](https://www.leads4pass.com/300-835.html)