```
import requests
url = "https://myDNAserver/dna/intent/api/v1/network-device"
payload = {}
headers = {'x-auth-token': 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1ZDE0OWZkMjhlZTY2
MmQ3NGM5YzE5ZTliLmYzMCIsImV4cCI6MTU3MjM3ODE5MCwidXNlcm5hbWUiOiJraaX
N371387657.zhK5LPQd501ZUpZl0IH_qrgOXttINbxSFFF7JOEtRIs'
}

response = requests.request("GET"            , url,
headers = headers                    , data = payload           )
print(response.text.encode('utf8'))
```

QUESTION 40
Refer to the exhibit.

A REST API retune this JSON output for a GET HTTP request, Which has assigned to a variable
called "vegetables" Using python, which output is the result of this command?

```
print(filter(lambda l: l['type'] == 'fruit', vegetables)[0]['items'][0]['items'][0])
```

A. Lettuce
B. Kiwi
C. ['kiwi', 'grape']
D. {'color': 'green', 'items': ['kiwi', 'grape']

Correct Answer: B

QUESTION 41
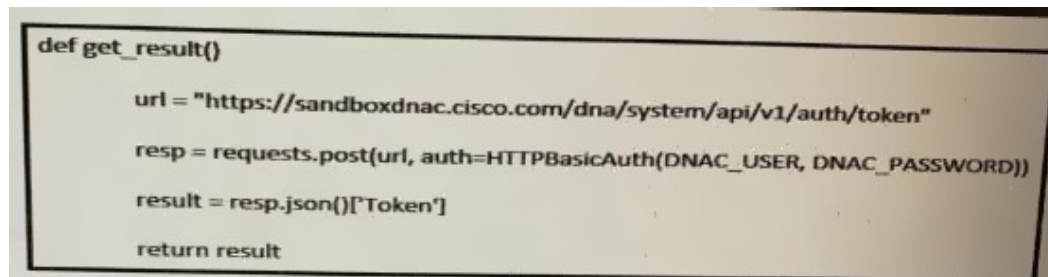On which port does NETCONF operate by default?

A. 23
B. 443
C. 822
D. 830

Correct Answer: D

QUESTION 42
Refer to the exhibit. What does the python function do?

```
def get_result()

        url = "https://sandboxdnac.cisco.com/dna/system/api/v1/auth/token"

        resp = requests.post(url, auth=HTTPBasicAuth(DNAC_USER, DNAC_PASSWORD))

        result = resp.json()['Token']

        return result
```
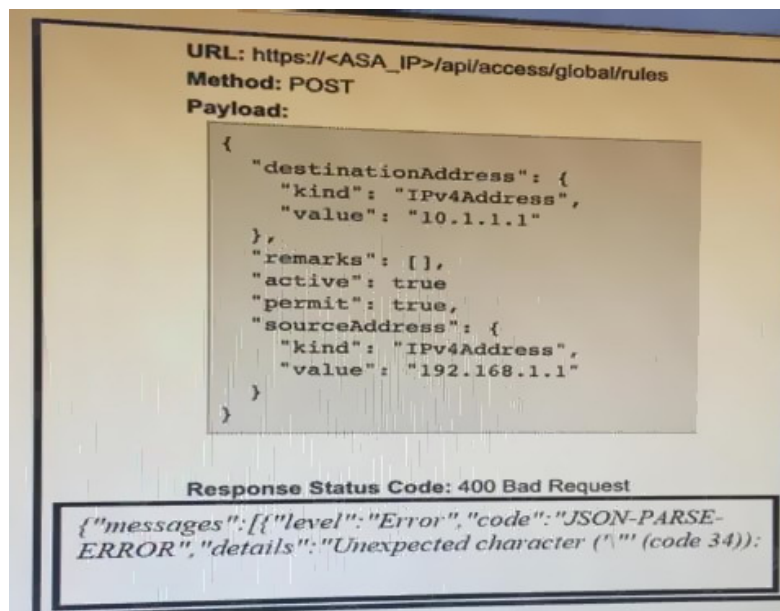
A. It returns DNAC user and password.
B. It returns HTTP Basic Authentication.
C. It returns an authorization token.
D. It reads a token from a local JSON file and posts the token to the DNAC URL.

Correct Answer: C

QUESTION 43
Refer to the exhibit. What caused the error in this API request?

```
URL: https://<ASA_IP>/api/access/global/rules
Method: POST
Payload:

{
    "destinationAddress": {
        "kind": "IPv4Address",
        "value": "10.1.1.1"
    },
    "remarks": [],
    "active": true,
    "permit": true,
    "sourceAddress": {
        "kind": "IPv4Address",
        "value": "192.168.1.1"
    }
}

Response Status Code: 400 Bad Request

{"messages":[{"level":"Error","code":"JSON-PARSE-
ERROR","details":"Unexpected character ('\"' (code 34)):
```

A. The API resource does not support the POST operation
B. The API resource does not support JSON format payloads.
C. The submitted JSON payload includes a field that is not supported by the API resource.
D. The submitted JSON payload has a formatting issue

Correct Answer: D

QUESTION 44
Refer to the exhibit. What is represented in this YANG module?

```
module ex-ethernet {
    namespace "http://example.com/ethernet";
    prefix "eth";
    import ietf-interfaces {
      prefix if;
    }
    augment "/if:interfaces/if:interface" {
      when "if:type = 'ethernetCsmacd'";
      container ethernet {
        must "../if:location" {
          description
            "An ethernet interface must specify the physical location of the ethernet hardware.";
        }
        choice transmission-params {
          case auto {
            leaf auto-negotiate {
              type empty;
            }
          }
          case manual {
            leaf duplex {
              type enumeration {
                enum "half";
                enum "full";
              }
            }
            leaf speed {
              type enumeration {
                enum "10Mb";
                enum "100Mb";
                enum "1Gb";
                enum "10Gb";
              }
            }
          }
        } // other ethernet specific params...
      }
    }
}
```

A. interface management
B. topology
C. BGP
D. OpenFlow

Correct Answer: A

QUESTION 45
Refer to the exhibit. A developer is troubleshooting an API with the given API with the given API documentation and cURL command. What is the cause of this problem?

List Messages

Lists all messages in a room. Each message will include content attachments if present.
The list sorts the messages in descending order by creation date.
Long result sets will be split into pages.

`GET /v1/messages`

Query Parameters

roomId
string Required
List messages in a room, by ID.

mentionedPeople
array
List messages with these people mentioned, by ID. Use me as a shorthand for the current API user.

before
string
List messages sent before a date and time.

beforeMessage
string
List messages sent before a message, by ID.

max
number
Limit the maximum number of messages in the response.
Default: 50

Header

Authorization
Use personal access token
Bearer ••••••••••••••••••••••••••••••
This limited-duration personal access token is hidden for your security.

Query Parameters

roomId
Required

mentionedPeople

before

beforeMessage

max

```
bash-3.2$ curl -H "Content-Type: application/json" -H "Authorization:
Fj2zzykEa0911c9GK2j8LtE1Hk\H6oRMPQdw1pAT60I7NDThhNwZl2B5pqMg14kK_b9ei59ISAC\Y7-
NarA-2n9M-tGgt-5xQ39iDejgcs" -i -X POST -d "{ \"roomId\":
\"Y2geK53sjELknosrC7SwQ5ZGL99pHgiuScB7DfNvUsr8Xx4wDKLiPORcEkryAnM3QmK9LQZsPOG4\"
, \"text\": \"test2\" }" https://api.ciscospark.com/v1/messages

HTTP/1.1 401 Unauthorized
Via: 1.1 linkerd
TrackingID: ROUTER_5E0FDC8C-EB6F-01BB-00EF-806BF1BD00EF
Date: Sat, 04 Jan 2020 00:30:04 GMT
Server: Redacted
Content-Length: 267
Content-Type: application/json
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload

{"message":"The request requires a valid access token set in the Authorization
request header.","errors":[{"description":"The request requires a valid access
token set in the Authorization request
header."}],"trackingId":"ROUTER_5E0FDC8C-EB6F-01BB-00EF-806BF1BD00EF"}

bash-3.2$
```
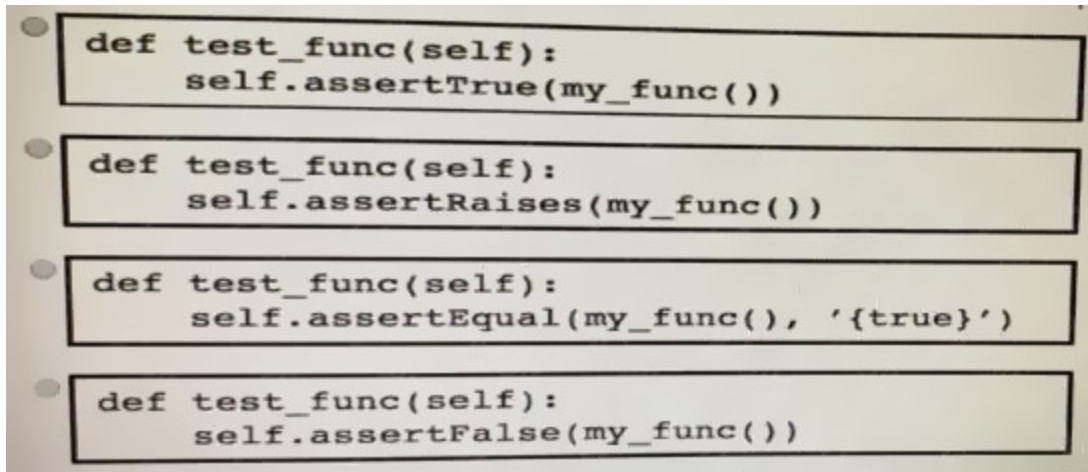
A. The user is not allowed to post massages from their account
B. The request body is missing or incomplete
C. The API token specified is expired
D. The authorization header is missing or incomplete

Correct Answer: C

QUESTION 46
A function my_func() returns True when it executes normally. Which python snippet tests my_func()?

```
def test_func(self):
    self.assertTrue(my_func())

def test_func(self):
    self.assertRaises(my_func())

def test_func(self):
    self.assertEqual(my_func(), '{true}')

def test_func(self):
    self.assertFalse(my_func())
```
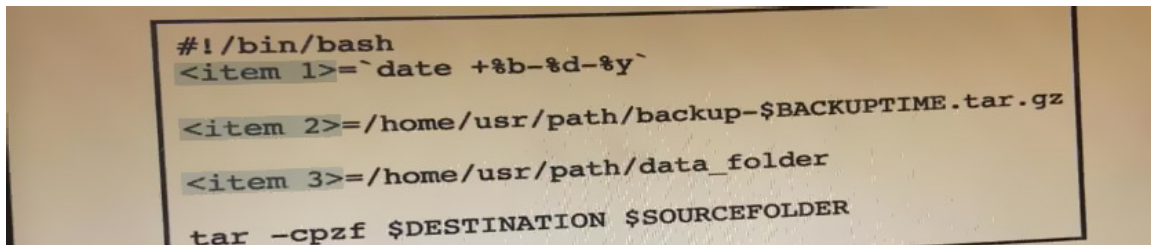
A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: B

QUESTION 47
DRAG DROP
Refer to the exhibit.

```
#!/bin/bash
<item 1>=`date +%b-%d-%y`

<item 2>=/home/usr/path/backup-$BACKUPTIME.tar.gz

<item 3>=/home/usr/path/data_folder

tar -cpzf $DESTINATION $SOURCEFOLDER
```

Drag and drop the variables from the left onto the item numbers on the right that match the missing assignments in the exhibit.

BACKUPTIME	<item 1>
SOURCEFOLDER	<item 2>
DESTINATION	<item 3>

Correct Answer: