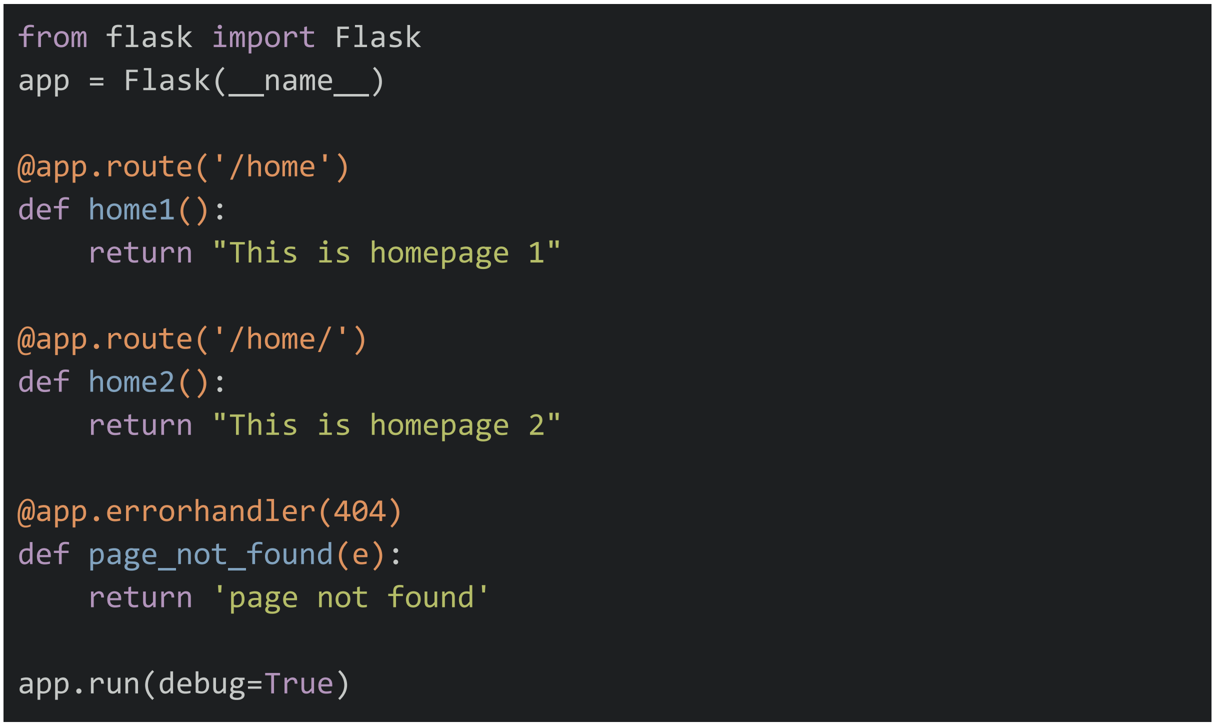
AppDevQuiz-II Jan24

Q1. Consider the following code.

[MCQ: 2 points]



```python
from flask import Flask
app = Flask(__name__)

@app.route('/home')
def home1():
    return "This is homepage 1"

@app.route('/home/')
def home2():
    return "This is homepage 2"

@app.errorhandler(404)
def page_not_found(e):
    return 'page not found'

app.run(debug=True)
```

If the flask application is running on http://127.0.0.1:5000, what will browser render for URL http://127.0.0.1:5000/home/

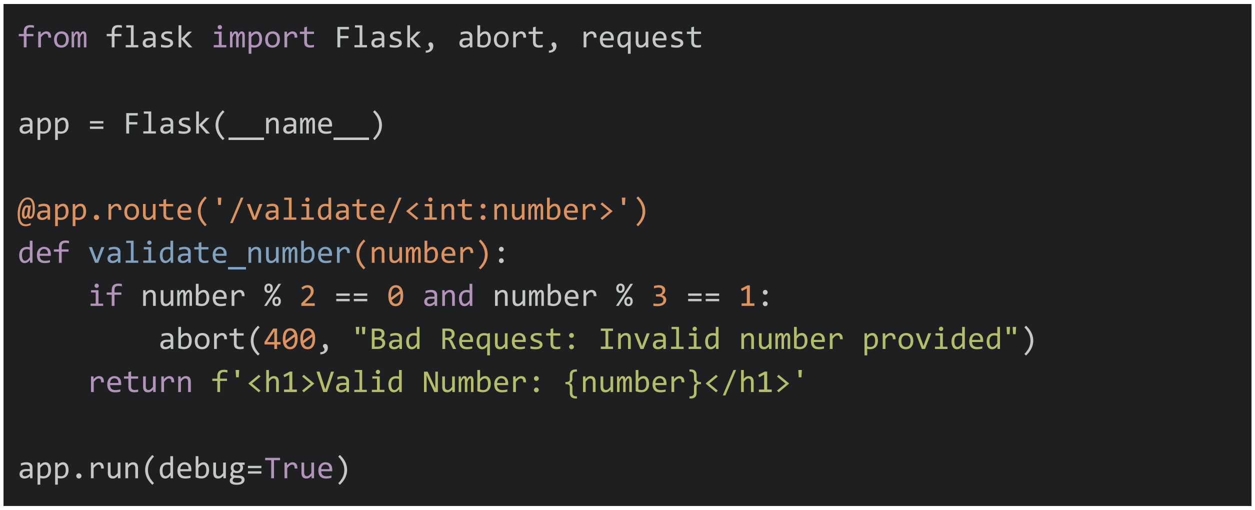
A. Page not found
B. This is homepage 1
C. This is homepage 2
D. Code will throw error

Answer: C

Q2. Consider the following flask application.

[MSQ: 3 points]

app.py



```
from flask import Flask, abort, request

app = Flask(__name__)

@app.route('/validate/<int:number>')
def validate_number(number):
    if number % 2 == 0 and number % 3 == 1:
        abort(400, "Bad Request: Invalid number provided")
    return f'<h1>Valid Number: {number}</h1>'

app.run(debug=True)
```

Which of the following statements is/are true if the application is running locally on http://127.0.0.1:5000 ?

A. For URL `http://127.0.0.1:5000/validate/4` the browser will render
   Valid Number: 4

B. For URL `http://127.0.0.1:5000/validate?number=5` the browser will render
   Bad Request: Invalid number provided

C. For URL `http://127.0.0.1:5000/validate/4` the browser will render
   Bad Request: Invalid number provided

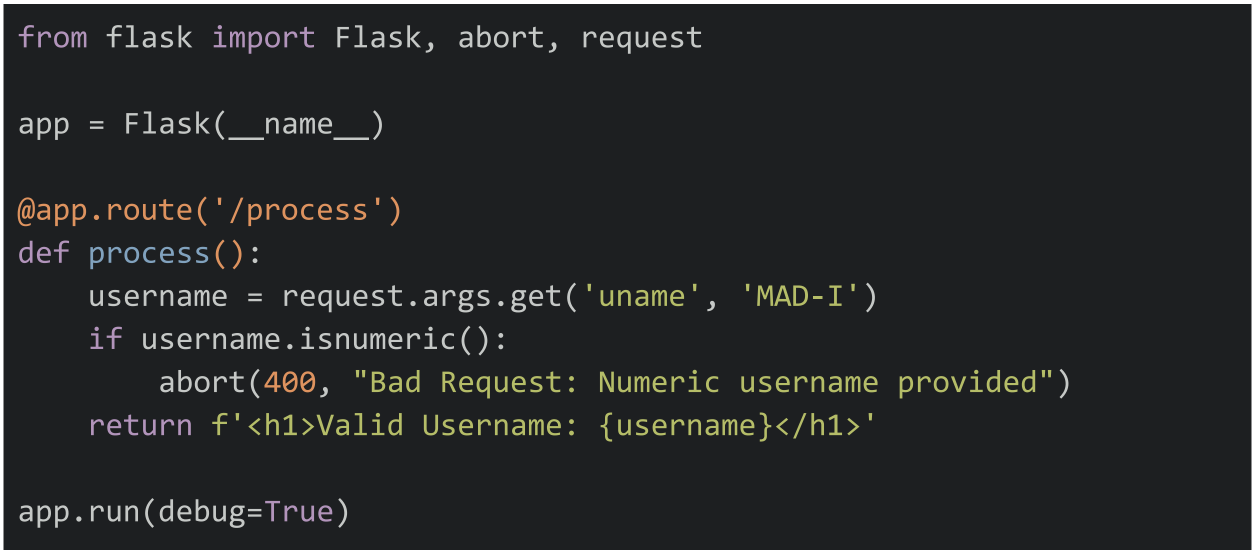
D. For URL `http://127.0.0.1:5000/validate/5` the browser will render
   Valid Number: 5

Answer: C, D

Q3. Consider the following flask application.

[MSQ: 3 points]

app.py



```python
from flask import Flask, abort, request

app = Flask(__name__)

@app.route('/process')
def process():
    username = request.args.get('uname', 'MAD-I')
    if username.isnumeric():
        abort(400, "Bad Request: Numeric username provided")
    return f'<h1>Valid Username: {username}</h1>'

app.run(debug=True)
```

Which of the following statements is/are true if the application is running locally on http://127.0.0.1:5000 ?

A. For URL `http://127.0.0.1:5000?uname=MAD-II` the browser will render
Valid Username: MAD-II

B. For URL `http://127.0.0.1:5000/process` the browser will render
Valid Username: MAD-I

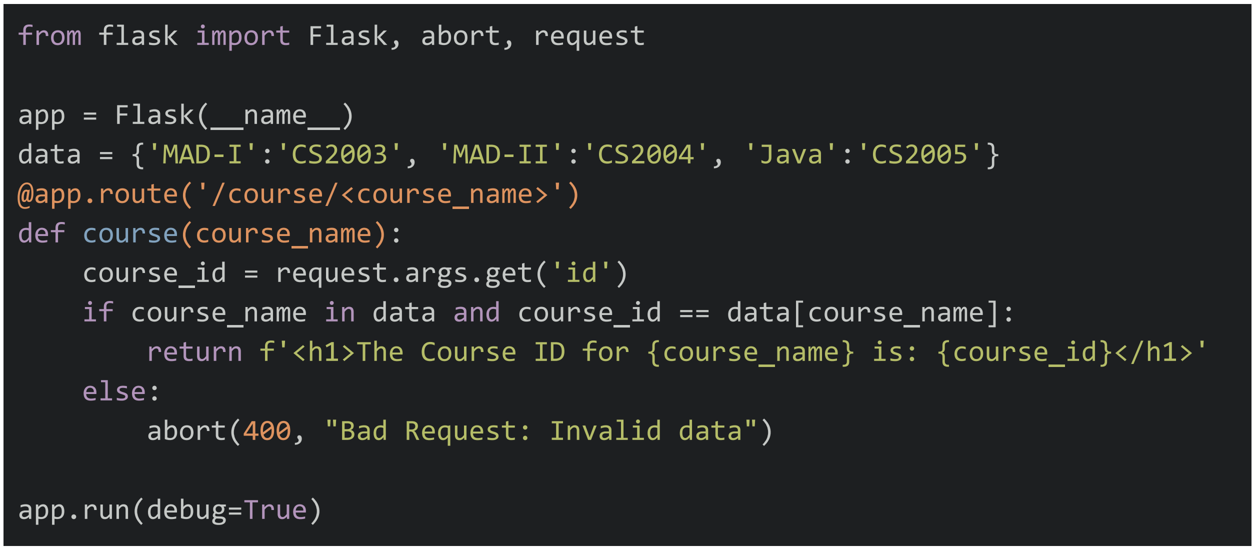
C. For URL `http://127.0.0.1:5000/process?uname` the browser will render
Valid Username:

D. For URL `http://127.0.0.1:5000/process?uname=101MU` the browser will render Bad Request: Numeric username provided

Answer: B, C

Q4. Consider the following flask application.

[MSQ: 3 points]
app.py



```
from flask import Flask, abort, request

app = Flask(__name__)
data = {'MAD-I':'CS2003', 'MAD-II':'CS2004', 'Java':'CS2005'}
@app.route('/course/<course_name>')
def course(course_name):
    course_id = request.args.get('id')
    if course_name in data and course_id == data[course_name]:
        return f'<h1>The Course ID for {course_name} is: {course_id}</h1>'
    else:
        abort(400, "Bad Request: Invalid data")

app.run(debug=True)
```

Which of the following statements is/are true if the application is running locally on http://127.0.0.1:5000 ?

A. For URL `http://127.0.0.1:5000/MAD-I?id=CS2003` the browser will render;
   The Course ID for MAD-I is: CS2003

B. For URL `http://127.0.0.1:5000/course/Java?id=CS2005` the browser will render;
   The Course ID for Java is: CS2005

C. For URL `http://127.0.0.1:5000/course?course_name=MAD-II&id=CS2004` the browser will render;
   The Course ID for MAD-II is: CS2004

D. For URL `http://127.0.0.1:5000/course/MAD-II?id=CS2003` the browser will render;
   Bad Request: Invalid data

Answer: B, D

Q5. Consider a character set M, which consists of only those characters used in the statement, "sphinx of black quartz judge my vow" If this statement is to be saved in a document with minimum encoding, what will be the size of the document given that no other information or context is to be saved?

[MCQ: 3 points]

A. 145 bits

B. 245 bits

C. 175 bits

D. 256 bits

Answer: C

Q6. Consider the following Python code snippet.

[MCQ: 3 points]

```python
from string import Template

text = "The $color car is parked in front of the $building."

template = Template(text)

print(=== OUTPUT ===)
```

Which of the following statements, when substituted in place of `=== OUTPUT ===`, will throw a KeyError?

A.

```python
template.substitute({'color':'red'})
```

B.

```python
template.substitute({'color':'blue','building':'office', 'place': 'Chennai'})
```

C.

```python
template.safe_substitute({'color':'green'})
```

D.

```python
template.safe_substitute({'color':'yellow','building':'apartment', 'place': 'Chennai'})
```

Answer: A

Q7. Software packages A and B of complexity O($nlogn$) and O(n^2), respectively, spend exactly T_A(n) = $C_A nlog_{10}(n)$ and T_B(n) =$C_B n^2$ milliseconds to process n data items. During a test, the average time of processing n = 10^7 data items with the package A and B is 420 milliseconds and 350 milliseconds, respectively. The time taken (in milliseconds) by software package A and B to process 10^8 data items, respectively will be?

[MCQ: 3 points]

A. 48 and 35
B. 480 and 350
C. 4800 and 35000
D. 48000 and 35000

Answer: C

Q8. Consider the following Flask code snippet.

[MCQ: 2 points]

```
from flask import Flask, request

app = Flask(__name__)
@app.route('/greet/<name>')
def greet(name):
    if name:
        return f'<h1>Hi, Mr/Ms/ {name} welcome to flask</h1>'
    else:
        return f'<h1>Hi, Mr/Ms/ <Unknown> welcome to flask</h1>'

app.run(debug=True)
```

If the application is running on a local server for the URL: https://127.0.0.1:5000, what will the browser render for the URL, http://127.0.0.1:5000/greet/?

A.

Hi, Mr/Ms/ None welcome to flask

B.

Hi, Mr/Ms/ <unknown> welcome to flask

C.

Bad request, Invalid name provided

D.

Not Found

Answer: D

Q9. Consider the below schema:

[MSQ: 4.5 points]

Field	Description
emp_id	Integer and primary key
employee_name	String with maximum 255 length, mandatory field and can be duplicated
basic_salary	Real number is an optional field that can be duplicated

Which of the below is the correct syntax to create a data model using `flask_sqlalchemy`?

A.

```
class Employee(db.Model):
    emp_id=db.Column(db.Integer, unique=True)
    employee_name=db.Column(db.String, nullable=False)
    basic_salary=db.Column(db.Float, nullable=True)
```

B.

```
class Employee(db.Model):
    emp_id=db.Column(db.Integer, primary_key=True)
    employee_name=db.Column(db.String, nullable=True)
    basic_salary=db.Column(db.Float, nullable=False)
```

C.

```
class Employee(db.Model):
    emp_id=db.Column(db.Integer, primary_key=True)
    employee_name=db.Column(db.String, nullable=False)
    basic_salary=db.Column(db.Float)
```

D.

```
class Employee(db.Model):
    emp_id=db.Column(db.Integer, primary_key=True)
    employee_name=db.Column(db.String, nullable=False)
    basic_salary=db.Column(db.Float, nullable=True)
```

Answer: C, D

Q10. Consider the “loans” table below, and it’s model class “Loans” associated with the table “loans” in the SQLite db.

[MCQ: 3 points]

ID	application_date	approved_status	Remarks
1	27/01/2023	Yes	All docs verified
2	29/03/2023	No	The mortgage doc is missing
3	13/04/2023	No	The payslip doc is not enclosed
4	03/08/2023	No	TDS doc is not enclosed
6	10/12/2023	Yes	All docs verified
7	17/12/2023	Yes	All docs verified

What is the correct output of the `flask_sqlalchemy` commands below?

```
loans=Loans.query.filter_by(approved_status= "No").all()
for l in loans:
     print(l.ID, ", ", l.approved_status, ", ", l.remarks)
```

A.

```
1, Yes, All docs verified
6, Yes, All docs verified
7, Yes, All docs verified
```

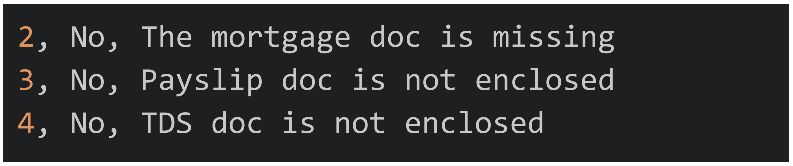
B.

```
2, No, All docs verified
3, No, All docs verified
4, No, All docs verified
```

C.

```
1, No, All docs verified
6, No, All docs verified
7, No, All docs verified
```

D.



```
2, No, The mortgage doc is missing
3, No, Payslip doc is not enclosed
4, No, TDS doc is not enclosed
```

Answer: D

Q11. Consider the below statements about Cookies:

[MCQ: 2 points]

Statement 1: Cookies enable the web app to recognize the users
Statement 2: Cookies can include information about location, login data, and language etc.,

Which of the below is the correct option?

A. Statement 1 is correct and Statement 2 is incorrect
B. Statement 1 is incorrect and Statement 2 is correct
C. Both Statement 1 and Statement 2 are correct
D. Both Statement 1 and Statement 2 are incorrect

Answer: C

Q12. Consider the below Flask code snippet:

[MCQ: 2 points]

```
from flask import Flask, jsonify
app = Flask(__name__)

@app.route('/api/courses')
def get_courses():
    courses = [{'id': 'CS1001', 'name' : 'CT'}, {'id' : 'CS1002',
                'name': 'PDSA'}]
    return jsonify(courses)
```

If the application is running locally, what will be the ***status*** code and ***mimetype*** of the response for the URL, `http:/127.0.0.1:5000/api/courses`?

A.

```
status=404 and mimetype='application/json'
```

B.

```
status=200 and mimetype='application/json'
```

C.

```
status=200 and mimetype='text/javascript'
```

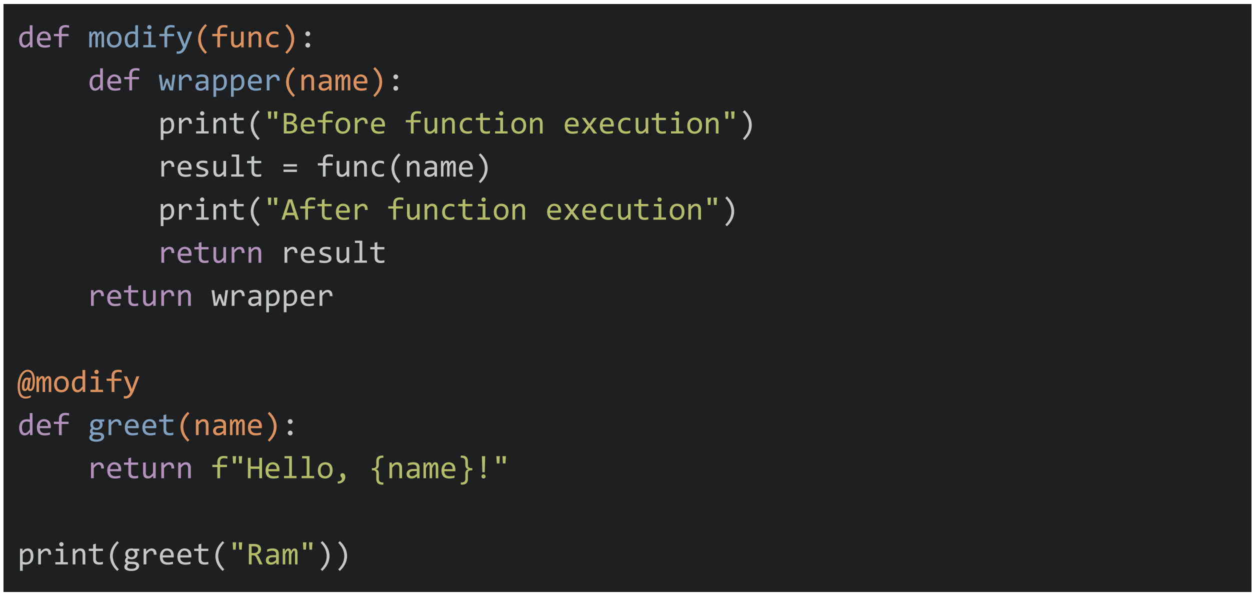
D.

```
status=404 and mimetype='text/javascript'
```

Answer: B

Q13. Consider the following Python code snippet.

[MCQ: 3 points]



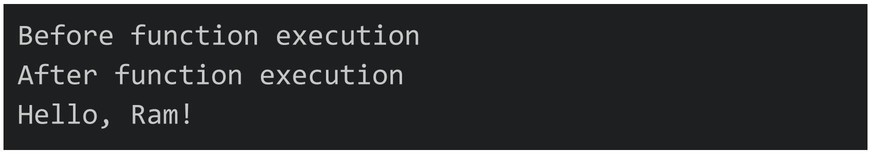
```python
def modify(func):
    def wrapper(name):
        print("Before function execution")
        result = func(name)
        print("After function execution")
        return result
    return wrapper

@modify
def greet(name):
    return f"Hello, {name}!"

print(greet("Ram"))
```

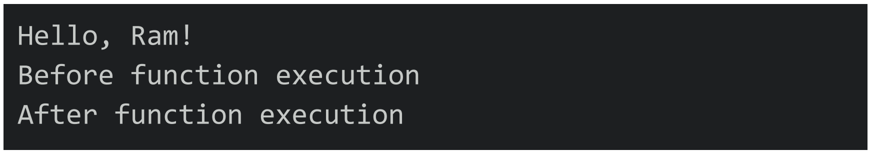
What will be the output of the code on the terminal?

A.



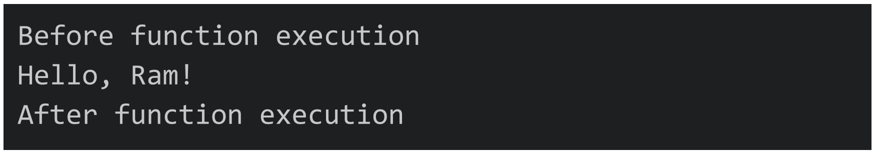
```
Before function execution
After function execution
Hello, Ram!
```

B.



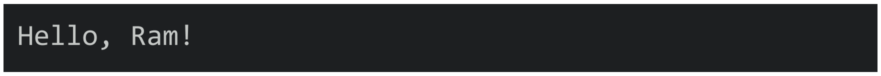
```
Hello, Ram!
Before function execution
After function execution
```

C.



```
Before function execution
Hello, Ram!
After function execution
```

D.



```
Hello, Ram!
```

Answer: Option A

Q14. Consider the following HTML document and select the correct option(s).

[MSQ: 4.5 points]

```
<!DOCTYPE html>
<html>
<head>
    <title>Form Validation</title>
    <script>
        function validateForm() {
            var username = document.forms["myForm"]["username"].value;
            var password = document.forms["myForm"]["password"].value;
            if (username == "") {
                alert("Username must be filled out");
                return false;
            }
            if (password.length < 8) {
                alert("Password must be at least 8 characters long");
                return false;
            }
        }
    </script>
</head>
<body>
    <form name="myForm" onsubmit="validateForm()">
        <label for="username">Username:</label>
        <input type="text" name="username">
        <label for="password">Password:</label>
        <input type="password" name="password" required>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

A. The JavaScript function, `'validateForm'` performs the backend processing of form data.

B. The JavaScript function, `'validateForm'` carries out the client side validation of form data.

C. If the password field is left empty, on submission, the user will get a prompt **"Please fill out this field."** near the password field.

D. If the password field is left empty, on submission, the user will get a prompt **"Password must be at least 8 characters long"** from the browser.

Answer: Option B and C.

Q15. Consider the following Python code snippet.

[MCQ: 4.5 points]

```
import logging
 import sys

logging.basicConfig(level=logging.INFO,
                    format='%(levelname)s - %(message)s')

num1, num2 = int(sys.argv[1]), int(sys.argv[2])

logging.info("This operation checks factors")

if num1 % num2 == 0:
    logging.debug(f"{num2} is a factor of {num1}")
else:
    logging.warning(f"{num2} is not the factor of {num1}")
```

What will be the output of the above code on the terminal for the command:

```
python log.py 25 5
```

A.

```
INFO - This operation checks factors
WARNING - 5 is not the factor of 25
```

B.

```
INFO - This operation checks factors
DEBUG - 5 is a factor of 25
```

C.

```
INFO - This operation checks factors
```

D.

```
DEBUG - 5 is a factor of 25
```

Answer: Option C

Q16. Consider the following api resources created using flask_restful. Assume that the app is connected to the database and is running on one terminal. Select the correct option(s).

[MSQ: 4.5 points]

```
class TestApi(Resource):
    def get(self, admin):
        # retrieves admin data from database and returns it.
        return {
            "name": "mad1_admin",
            "role": "admin",
        }

    def post(self):
        # fetches user data from request body and stores in the
        database.
        return {
            "message": "user added successfully"
        }, 201

api.add_resource(TestApi, "/api/<string:admin>", "/<admin>",
"/user_data", "/")

app.run()
```

A. The base URL for the app is mapped with GET HTTP method.

B. If we run the command: `curl http://127.0.0.1:5000/api/user -X GET` on a new terminal; it will return;

```
{
  "name": "mad1_admin",
  "role": "admin"
}
```

C. The endpoint, `/<admin>` is mapped with GET HTTP method.

D. If we run the command: `curl http://127.0.0.1:5000/user_data -X POST` on a new terminal; it will return;

```
{
  "message": "user added successfully"
}
```

Answer: B, C and D