

Consider the following recurrences and choose the **correct** option.

1. $T_1(n) = 9T_1(n/3) + O(n)$

Base Case : $T_1(1) = O(1)$

2. $T_2(n) = 3T_2(n/4) + O(n^2)$

Base Case : $T_2(1) = O(1)$

Options :

6406531484601. ✖ $T_1 = O(n^{\log_2 3})$ and $T_2 = O(n^2)$

6406531484602. ✔ $T_1 = O(n^2)$ and $T_2 = O(n^2)$

6406531484603. ✖ $T_1 = O(n)$ and $T_2 = O(n^2)$

6406531484604. ✖ $T_1 = O(n^2)$ and $T_2 = O(n^{\log_3 4})$

AppDev1

Section Id :	64065328981
Section Number :	7
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	19
Number of Questions to be attempted :	19
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1

Sub-Section Id :

64065363318

Question Shuffling Allowed :

No

Is Section Default? :

null

Question Number : 119

Question Id : 640653445561

Question Type : MCQ

Is Question Mandatory : No

Calculator : None

Response Time : N.A

Think Time : N.A

Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL: MODERN APPLICATION DEVELOPMENT 1"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531484610.  Yes

6406531484611.  No

Sub-Section Number :

2

Sub-Section Id :

64065363319

Question Shuffling Allowed :

Yes

Is Section Default? :

null

Question Number : 120

Question Id : 640653445562

Question Type : MSQ

Is Question Mandatory : No

Calculator : None

Response Time : N.A

Think Time : N.A

Minimum Instruction Time : 0

Correct Marks : 3

Selectable Option : 0

Question Label : Multiple Select Question

Suppose a request "<https://xyz.com?name=Abhi&age=22>" generates the below response on the terminal.

```
Name : Abhi
```

The definition of the flask endpoint which handles the above request is given below,

```
@app.route(code1)
def getData():
    data = code2
    print("Name :", data)
    return "Hello World"
```

Which of the following options should be used to fill the placeholders "code1" and "code2", to achieve the desired result as shown above?

Options :

6406531484612. ✓ Code1: "/", methods = ['GET', 'POST']
Code2: request.args['name']

6406531484613. ✗ Code1: "/", methods = ['GET']
Code2: request.form['name']

6406531484614. ✓ Code1: "/"
Code2: request.args.get('name')

6406531484615. ✗ Code1: "/"
Code2: request.form['name']

Question Number : 121 Question Id : 640653445563 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following statement(s) is/are true?

Options :

6406531484616. ✓ The flask framework uses server side rendering (SSR) to deliver web pages.

6406531484617. ✓ In general, the HTTP GET request does not have a request body.

6406531484618. ✖ The HTML forms can be used to make "GET" and "DELETE" HTTP requests.

6406531484619. ✖ All of these

Question Number : 122 Question Id : 640653445572 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

Consider the following model classes "Section" and "Book" corresponding to tables "section" and "book" respectively in the SQLite database.

```
class Section(db.Model):
    section_id = db.Column(db.Integer(), primary_key = True)
    section_name = db.Column(db.String(50), nullable = False)
    books = db.relationship("Book", backref = "sectionof")

class Book(db.Model):
    book_id = db.Column(db.Integer(), primary_key = True)
    book_name = db.Column(db.String(50), nullable = False)
    section = db.Column(db.Integer(), db.ForeignKey("section.section_id"))
```

If an object "s1" that represents an existing record in the table "section" is defined as s1 = Section.query.get(1), The correct way(s) to add a book with the name "Wings of fire" that belongs to s1 using the Python console is/are.

Options :

```
>>> b1 = Book(book_name = "Wings of fire", section = 1)
>>> db.session.add(b1)
6406531484652. ✓ >>> db.session.commit()
```

```
>>> b1 = Book(book_name = "Wings of fire", sectionof = 1)
>>> db.session.add(b1)
>>> db.session.commit()
```

6406531484653. ✖

6406531484654. ✖

```
>>> b1 = Book(book_name = "Wings of fire", section = s1)
>>> db.session.add(b1)
>>> db.session.commit()
```

```
>>> b1 = Book(book_name = "Wings of fire", sectionof = s1)
>>> db.session.add(b1)
>>> db.session.commit()
```

6406531484655. ✓

Question Number : 123 Question Id : 640653445579 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following statements is/are true about indexing?

Options :

6406531484680. ✓ Indexes are special lookup tables that the database search engine can use to speed up data retrieval.

6406531484681. ✓ Indexes can be created or dropped without any effect on the data.

6406531484682. ✓ Indexes should be avoided for tables that have frequent insert operations.

6406531484683. ✖ Indexes enhance the performance even if the table is updated frequently.

Sub-Section Number : 3

Sub-Section Id : 64065363320

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 124 Question Id : 640653445564 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following SQLALchemy class models.

```
class College(Base):
    __tablename__ = "college"
    college_id = Column(Integer, primary_key=True)
    college_name = Column(String(50))
    students = relationship("Student", back_populates="college",
uselist=False)

class Student(Base):
    __tablename__ = "student"
    student_id = Column(Integer, primary_key=True)
    student_name = Column(String(30))
    parent_id = Column(Integer, ForeignKey("college.id"), unique=True)
    college = relationship("College", back_populates="students")
```

Which of the following relationships is depicted between the “College” and “Student” models?

Options :

6406531484620. ✖ Many-to-Many

6406531484621. ✖ Many-to-One

6406531484622. ✖ One-to-Many

6406531484623. ✔ One-to-One

Question Number : 125 Question Id : 640653445565 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following python functions.

```
def module1(a, b, c):  
    mod1 = a%b+c  
    print(mod1)  
def module3(original_func):  
    def module2(*args):  
        module1(*args)  
        x,y,z = args  
        print((z)**2)  
    return module2(4,5,7)  
module3(module1)
```

Supposing the statement “module3(module1)” works in the way a decorator works, then what will be the output of the above python snippet.

Options :

6406531484624. ✖ 12

49

6406531484625. ✔ 11

49

6406531484626. ✖ 11

25

6406531484627. ✖ Error

Question Number : 126 Question Id : 640653445566 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following restful API implementation using Flask.

```
from flask import Flask, jsonify, request
from flask_restful import Resource, Api

app = Flask(__name__)
api = Api(app)
food_items = {"rice": "50", "pulses": "60", "wheat": "70"}

class Home(Resource):
    def get(self):
        return food_items

    def post(self):
        data=request.json
        food_items.update(data)
        return "Data is Inserted"

api.add_resource(Home, '/')

if __name__ == '__main__':
    app.run(debug = True)
```

If the above flask application is running on URL "<http://127.0.0.1:5000>", choose its output when we send POST and GET requests in the sequence given below.

1. POST "<http://127.0.0.1:5000>" with request body {"grape": "30"}
2. GET "<http://127.0.0.1:5000>"

Options :

1. "Data is Inserted"

2. {
 "rice": "50",
 "pulses": "60",
 "wheat": "70"
 "grape": "30"
}

6406531484628. ✓

6406531484629. ✗


```
1. {  
    "rice": "50",  
    "pulses": "60",  
    "wheat": "70"  
}
```

2. "Data is Inserted"

1. Error

```
2. {  
    "rice": "50",  
    "pulses": "60",  
    "wheat": "70"  
}
```

6406531484630. ✖

1. "Data is Inserted"

2. Error

6406531484631. ✖

Question Number : 127 Question Id : 640653445569 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the below flask application.

```
from flask_sqlalchemy import SQLAlchemy
from flask import Flask

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///testdb.sqlite3'
db = SQLAlchemy(app)

class student(db.Model):
    roll_no = db.Column('roll_no', db.Integer, primary_key = True)
    name = db.Column('name', db.String(100), unique = True)

    def __init__(self, roll_no, name):
        self.roll_no = roll_no
        self.name = name

db.create_all()
student1 = student(roll_no = 1, name = 'Sam')
db.session.add(student1)
db.session.commit()
student2 = student(roll_no = 2, name = 'Lee')
student3 = student(roll_no = 3, name = 'Sandy')
db.session.add(student2)
db.session.commit()
db.session.add(student3)

result = student.query.all()
print([x.name for x in result])
```

If you run the flask application using a terminal. What will be the output in the terminal?

Options :

6406531484640. ✖ ['Sam', 'Lee'] will be displayed in the terminal and two tuples will be added in the student relation in the "testdb" database.

6406531484641. ✔ ['Sam', 'Lee', 'Sandy'] will be displayed in the terminal and two tuples will be added in the student relation in the "testdb" database.

6406531484642. ✖ ['Sam', 'Lee', 'Sandy'] will be displayed in the terminal and three tuples will be added in the student relation in the "testdb" database.

6406531484643. ✖ [] will be displayed in the terminal and no tuple will be added in the student relation in the "testdb" database.

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

A search function S with number of inputs 'N' varies with time 't' according to relation $N = \log_{10} t$. What is the nature of algorithmic complexity of the search function S ?

Options :

6406531484660. ✖ Logarithmic

6406531484661. ✖ Cubic

6406531484662. ✔ Exponential

6406531484663. ✖ Bi-Quadratic

Sub-Section Number :	4
Sub-Section Id :	64065363321
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 129 Question Id : 640653445567 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4.5

Question Label : Multiple Choice Question

Consider the files given below.

File 1: macros.py

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

@app.route('/')
@app.route('/home')
def home():
    return render_template('index.html', my_list = [1, 5, 10, 69, 25, 30, 34, 45,])

app.run(debug=True)
```

File 2: index.html

```
{% from "macro.html" import calc %}
<!DOCTYPE html>
<html>
  <body>
    {{ calc(my_list) }}
  </body>
</html>
```

File 3: macro.html

```
{% macro calc(my_list) %}
  <p> The original list :{{ my_list }} </p>
  The modified list is:
  {% for x in my_list %}
    {% if (x % 2) == 0 and (x % 3) == 0 %}
      <p>{{x}}</p>
    {% elif (x % 5) == 0 %}
      {{x}}
    {% endif %}
  {% endfor %}
{% endmacro %}
```

What will be rendered by the browser if we hit the URL " <http://127.0.0.1:5000/>" or "<http://127.0.0.1:5000/home>"?

Options :

6406531484632. ✖ The original list :[1, 5, 10, 69, 25, 30, 34, 45]
The modified list is: 5 10 25 30 45

6406531484633. ✔ The original list :[1, 5, 10, 69, 25, 30, 34, 45]
The modified list is: 5 10 25
30
45

The original list :[1, 5, 10, 69, 25, 30, 34, 45]

The modified list is:

5

10

25

30

6406531484634. ✖ 45

The original list :[1, 5, 10, 69, 25, 30, 34, 45]

6406531484635. ✖ The modified list is: 5 10 69 25 30 34 45

Sub-Section Number : 5

Sub-Section Id : 64065363322

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 130 Question Id : 640653445568 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the below flask application.

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

@app.route('/hello/<int:a>/<int:b>', methods = ['GET', 'POST'])
def doOp1(a,b):
    return a+b

if __name__ == '__main__':
    app.run(debug = True)
```

If the application is running on “http://127.0.0.1:5000” and a request is sent to the URL “<http://127.0.0.1:5000/hello/2/1>”. What will be rendered by the browser?

Options :

6406531484636. ✖ 2

6406531484637. ✖ 12

6406531484638. ✔ 3

6406531484639. ✖ 21

Question Number : 131 Question Id : 640653445573 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following code block.

```
from flask import Flask, request

app = Flask(__name__)

#===== START =====

    Code Here

#===== END =====

app.run(debug = True)
```

Which of the following controllers will come in place of “Code Here” that can handle the given request URL below:

<http://127.0.0.1:5000/params?course=MAD1&stream=programming>

Options :

6406531484656. ✖

```
@app.route('/')
def info():
    params = request.args
    return f"<h2>The course {params['course']} belongs to {params['stream']} stream </h2>"
```
6406531484657. ✔

```
@app.route('/params')
def info():
    arg = request.args
    return f"<h2>The course {arg['course']} belongs to {arg['stream']} stream </h2>"
```
6406531484658. ✖

```
@app.route('/params/<course>/<stream>')
def info(course, stream):
    return f"<h2>The course {course} belongs to {stream} stream </h2>"
```
6406531484659. ✖

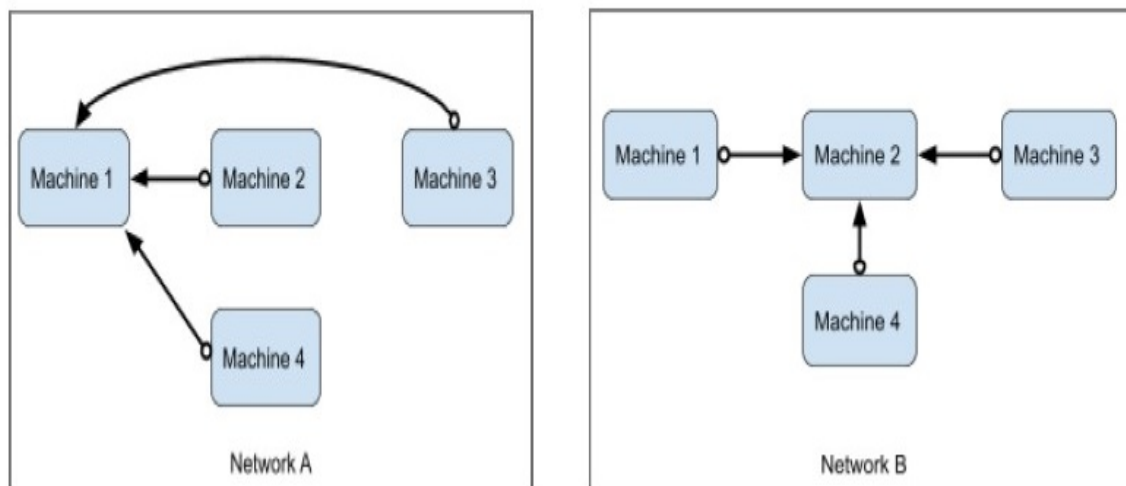
```
@app.route('/<course>/<stream>')
def info(course, stream):
    return f"<h2>The course {course} belongs to {stream} stream </h2>"
```

Question Number : 132 Question Id : 640653445575 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider two network structures given below. If a request can transmit only from Dot (o) to arrow (→) and responses from arrow (→) to Dot (o), Which of the following statements is correct?



Options :

6406531484664. ✖ Network A is a client-server model, whereas Network B is a peer-to-peer model.

6406531484665. ✖ Network B is a client-server model, whereas Network A is a peer-to-peer model.

6406531484666. ✔ Both the networks A and B represent client-server.

6406531484667. ✖ Both the networks A and B represent peer-to-peer.

Question Number : 133 Question Id : 640653445576 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following operations is not idempotent, in the context of REST?

Options :

6406531484668. ✖ GET

6406531484669. ✓ POST

6406531484670. ✖ PUT

6406531484671. ✖ DELETE

Question Number : 134 Question Id : 640653445577 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following is/are the advantage(s) of RESTful web service, being stateless?

A: Web services can treat each method request independently.

B: Web services need not maintain the client's previous interactions. It simplifies application design.

C: As HTTP is itself a stateless protocol, RESTful Web services work seamlessly with HTTP protocol.

Options :

6406531484672. ✖ Only A and B

6406531484673. ✖ Only A and C

6406531484674. ✖ Only B and C

6406531484675. ✓ All A, B and C

Question Number : 135 Question Id : 640653445578 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following column names for the table "Person":

Person

roll_no	name	father_name	DOB
---------	------	-------------	-----

Which of the following is the correct query syntax in SQLite to create UNIQUE index with name student_multiindex on multiple column (name, father_name and DOB) for table student_info?

Options :

6406531484676. ✖

```
CREATE UNIQUE INDEX student_multiindex ON Person  
COLUMN(name,father_name,DOB);
```

6406531484677. ✖

```
CREATE INDEX student_multiindex ON UNIQUE Person  
COLUMN(name,father_name,DOB);
```

6406531484678. ✖

```
CREATE INDEX student_multiindex ON Person  
COLUMN(name,father_name,DOB);
```

6406531484679. ✔

```
CREATE UNIQUE INDEX student_multiindex ON  
Person(name,father_name,DOB);
```

Sub-Section Number :

6

Sub-Section Id :

64065363323

Question Shuffling Allowed :

Yes

Is Section Default? :

null

Question Number : 136 Question Id : 640653445570 Question Type : MSQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following is/are true regarding DOM?

Options :

- 6406531484644. ✖ DOM stands for Document Object Module.
- 6406531484645. ✖ DOM is an API that is used to send the requests to and receive the responses from the server.
- 6406531484646. ✔ DOM is like a tree structure model of the web document.
- 6406531484647. ✔ Objects of the DOM can be manipulated with the help of javascript.

Sub-Section Number : 7
Sub-Section Id : 64065363324
Question Shuffling Allowed : Yes
Is Section Default? : null

Question Number : 137 Question Id : 640653445571 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0
Correct Marks : 4.5 Selectable Option : 0

Question Label : Multiple Select Question
 Suppose, Mr. Arun wants UTF-8 encoding of a document with 1000 characters with the given below image. Assuming, the document contains only one character being repeated one thousand times. The code point for the only character used 1000 times is U+00D8. What will be the size of the document?

1st Byte	2nd Byte	3rd Byte	4th Byte	Free Bits	Maximum Expressible Unicode Value
0xxxxxxx				7	007F hex (127)
110xxxxx	10xxxxxx			(5+6)=11	07FF hex (2047)
1110xxxx	10xxxxxx	10xxxxxx		(4+6+6)=16	FFFF hex (65535)
11110xxx	10xxxxxx	10xxxxxx	10xxxxxx	(3+6+6+6)=21	10FFFF hex (1,114,111)

- Options :**
- 6406531484648. ✔ 2000 bytes
 - 6406531484649. ✖ 8000 bytes
 - 6406531484650. ✔ 16000 bits
 - 6406531484651. ✖ 24000 bits

MLF

Section Id :	64065328982
Section Number :	8
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	14
Number of Questions to be attempted :	14
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065363325
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 138 Question Id : 640653445580 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL: MACHINE LEARNING FOUNDATIONS"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :