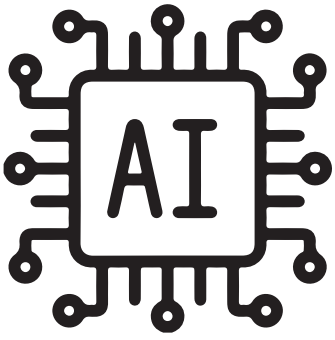
2024 TO 2026

SUBJECT & CHAPTER WISE SORTED & SOLVED
PREVIOUS YEAR QUESTIONS

GATE 2027

DATA SCIENCE AND ARTIFICIAL INTELLIGENCE



GATE 2027

Copyright 2026, IFAS Publications

All rights reserved. No part of this publication or the information contained herein may be reproduced, adapted, abridged, translated, stored in a retrieval system, computer system, photographic or other systems or transmitted in any form or by any means, electronic, mechanical, optical, digital, by photocopying, recording or otherwise, without written prior permission from the publisher. Any breach will attract legal action and prosecution without further notice.

Disclaimer: While every effort has been made to avoid errors and omissions, this publication is being sold and marketed on the understanding and presumption that neither the editors (or authors) nor the publishers nor the printers would be liable in any manner whatsoever, to any person either for an error or for an omission in this publication, or for any action to be taken on the basis of this work. Any inadvertent discrepancy noted may be brought to the attention of the publisher, for rectifying it in future editions, if published.

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the editors and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The editors and publisher have attempted to trace and acknowledge the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission and acknowledgement to publish in this form have not been obtained. If any copyright material has not been acknowledged please write and let us know so that we may rectify it.

Trademark Notice: Publications or corporate names may be trademarks, and are used only for identification and explanation in bonafide intent without intent to infringe.

Design & Published by:
IFAS Publications

Address:
Shivranjan Complex,
Someshwar Wadi, Near
Rajwada Hotel, Baner Road,
Pune 411 008.

Contact No.: **+91-960 489 7388**

Website: **https://ifasonline.com**

Edition: **2026**

Price: **₹ 895.00**

ISBN: **978-81-988636-2-1**

Preface

This book has been meticulously crafted to serve as a comprehensive guide for aspirants preparing for the **GATE Data Science and Artificial Intelligence examination**. With an ever-increasing competition, success in GATE demands a deep understanding of core concepts, coupled with a focused and strategic approach. This book has been structured to provide an all-encompassing review of essential topics while catering to the unique needs of both beginners and advanced learners.

This book is a comprehensive collection of **explanations of the previous year questions from the year 2024 to 2026**. Designed for aspirants who seek a deep understanding of question patterns and solution techniques, this book is an essential resource for efficient and effective exam preparation.

Organized into **eight chapters**, this collection covers the entire spectrum of the syllabus. **Each chapter is subdivided into units** for clear topic segmentation, **with questions sorted year-wise** to highlight the progression of concepts and formats over time. Additionally, **questions are categorized into Multiple Choice Questions (MCQ), Multiple Select Questions (MSQ), and Numerical Answer Type (NAT) formats**, reflecting the varied question styles found in the exam. **Practice questions are included at the end of every unit to help strengthen your understanding.** Each solution is presented in a detailed, step-by-step format, clarifying not only the approach but also the underlying concepts and problem-solving strategies required to succeed in GATE.

By working through these solutions, students can solidify their grasp of key topics, identify trends in question formats, and refine their time management skills. This book is intended to be a practical tool for achieving a high level of preparation, enabling candidates to approach the GATE exam with confidence and competence.

About GATE

Graduate Aptitude Test in Engineering (GATE) is a national examination conducted jointly by the Indian Institute of Science (IISc), Bengaluru and the seven Indian Institutes of Technology (at Bombay, Delhi, Guwahati, Kanpur, Kharagpur, Madras and Roorkee) on behalf of the National Coordination Board (NCB)-GATE, Department of Higher Education, Ministry of Education (MoE), Government of India.

Qualifying in GATE is a mandatory requirement for seeking admission and/or financial assistance to: (i) Master's programs and direct Doctoral programs in Engineering/ Technology/ Architecture and (ii) Doctoral programs in relevant branches of Arts and Science, in the institutions supported by the MoE and other Government agencies. Even in some colleges and institutions, which admit students without MoE scholarship/ assistantship, the GATE qualification is mandatory. Further, many Public Sector Undertakings (PSUs) have been using the GATE score in their recruitment process.

Exam Details:

Mode of Examination	Computer Based Test
Paper Code	DA
Duration of Exam	180 minutes
Total number of Questions	65
Total Marks	100
Language of Examination	English

Exam Pattern:

Sections	**Section I** (General Aptitude)	**Section II**
Number of Questions	10	55
Marks	5 questions of 1 mark each and 5 questions of 2 marks each	25 questions of 1 mark each and 30 questions of 2 marks each
Negative Marking	For 1-mark MCQ, 1/3 mark will be deducted for a wrong answer. For 2-mark MCQ, 2/3 mark will be deducted for a wrong answer. There will be no negative marking for MSQ & NAT.	
Types of Question	MCQ	MCQ, MSQ, NAT

Duration and Examination Type

All the papers of the GATE 2027 examination will be for 3 hours duration and they consist of 65 questions for a total of 100 marks. Since the examination is an ONLINE computer-based test (CBT), at the end of the stipulated time (3 hours), the computer screen will automatically close the examination inhibiting any further action.

Candidates will be permitted to occupy their allotted seats 40 minutes before the scheduled start of the examination. Candidates can login and start reading the instructions 20 minutes before the start of examination. The late login time (if any) recorded by the computer system MUST NOT be beyond 30 minutes from the actual examination starting time. Candidates will NOT be permitted to login after 30 minutes from the actual examination starting time. Candidates will NOT be permitted to leave the examination hall before the end of the examination.

Pattern of Question Papers

GATE 2027 may contain questions of THREE different types in all the papers:

(i) **Multiple Choice Questions (MCQ)** carrying 1 or 2 marks each, in all the papers and sections. These questions are objective in nature, and each will have a choice of four answers, out of which ONLY ONE choice is correct.

Negative Marking for Wrong Answers: For a wrong answer chosen in a MCQ, there will be negative marking. For 1-mark MCQ, 1/3 mark will be deducted for a wrong answer. Likewise, for 2-mark MCQ, 2/3 marks will be deducted for a wrong answer.

(ii) **Multiple Select Questions (MSQ)** carrying 1 or 2 marks each in all the papers and sections. These questions are objective in nature, and each will have choice of four answers, out of which ONE OR MORE than one choice(s) are/correct.

Note: There is NO negative marking for a wrong answer in MSQ questions. However, there is NO partial credit for choosing partially correct combinations of choices or any single wrong choice.

(iii) **Numerical Answer Type (NAT)** Questions carrying 1 or 2 marks each in most of the papers and sections. For these questions, the answer is a signed real number, which needs to be entered by the candidate using the virtual numeric keypad on the monitor (keyboard of the computer will be disabled). No choices will be shown for these types of questions. The answer can be a number such as 10 or −10 (an integer only). The answer may be in decimals as well, for example, 10.1 (one decimal) or 10.01 (two decimals) or −10.001 (three decimals). These questions will be mentioned with, up to which decimal places, the candidates need to present the answer. Also, for some NAT type problems an appropriate range will be considered while evaluating these questions so that the candidate is not unduly penalized due to the usual round-off errors. Candidates are advised to do the rounding off at the end of the calculation (not in between steps). Wherever required and possible, it is better to give NAT answer up to a maximum of three decimal places.

Note: There is NO negative marking for a wrong answer in NAT questions. Also, there is NO partial credit in NAT questions.

Scholarship/ Assistantship for Postgraduate Programs

To avail the financial assistance (scholarship/ assistantship) for a postgraduate program, the candidate must first secure the admission to a program in one of the centrally funded institutes, by a procedure that could vary from institute to institute. Depending upon the norms adopted by a specific institute or department of the institute, candidates may be admitted directly into a program based on their performance in GATE only; or based on their performance in GATE and an admission test and/or interview conducted by the department to which they have applied and/or the candidate's academic record. In the test/ interview-based selection procedure, as per the MoE guidelines, a minimum of 70% weightage will be given to the performance in GATE and the remaining will be given to the candidate's performance in test/ interview and/or academic record. However, the admitting institutes could prescribe a minimum passing mark in the test/ interview.

Candidates are advised to seek the complete details of the admission procedures and the availability of MoE scholarship/assistantship from the corresponding admitting institutions. The criteria for postgraduate admission with scholarship/ assistantship varies from one institute to the other. The management of the post-graduate scholarship/ assistantship is also the responsibility of the admitting institute. Similarly, reservation of the seats for different categories will be as per the policies and norms of the admitting institution and the Government of India rules.

The admitting institute may also specify the number of candidates who will be provided financial assistance (scholarship), if admission is secured. Qualification in GATE is also a minimum requirement to apply for various fellowships awarded by many Government organizations.

Different types of scholarships and sponsorships opportunities also exist for foreign nationals. Details may be obtained from nearby Indian Mission or from the website of Indian Council for Cultural Relations.

Use of GATE Score for Employment

In the past, several Public Sector Undertakings (PSUs) have used GATE score to shortlist the candidates for employment. A few such organizations are: Airports Authority of India (AAI), Bharat Heavy Electricals Limited (BHEL), Bharat Sanchar Nigam Limited (BSNL), Coal India Limited (CIL), Centre for Railway Information Systems (CRIS), Chenab Valley Power Projects Limited (CVPPL), Damodar Valley Corporation (DVC), Electronics Corporation of India Limited (ECIL), Engineers India Limited (EIL), Gas Authority of India Limited (GAIL), Hindustan Aeronautics Limited (HAL), Indian Oil Corporation Limited (IOCL), Mazagaon Dock Shipbuilders Limited (MDSL), National Aluminium Company Limited (NALCO), National Highways Authority of India (NHAI), NLC India Limited (NLCIL), National Mineral Development Corporation (NMDC), Nuclear Power Corporation of India Limited (NPCIL), National Thermal Power Corporation (NTPC), Oil and Natural Gas Corporation (ONGC), Odisha Power Generation Corporation Limited (OPGC), Power Grid Corporation of India Limited (POWERGRID), Power System Operation Corporation Limited (POSOCO), Rashtriya Ispat Nigam Limited (RINL), Steel Authority of India Limited (SAIL), etc.

Direct recruitment to Group-A level posts in central government, i.e., Senior Field Officer (Tele), Senior Research Officer (Crypto) and Senior Research Officer (S&T) in Cabinet Secretariat, Government of India, is now being carried out on the basis of GATE score. The details of the scheme of recruitment are normally published in National Newspapers/ Employment News/ Rozgar Samachar by the concerned authority.

Some other Government of India Organisations may also utilize GATE 2026 score for their recruitment purpose.

GATE Score

After the evaluation of the answers, the actual (raw) marks obtained by a candidate will be considered for computing the GATE Score. For multi-session test papers, raw marks obtained by the candidates in different sessions will be converted to Normalized marks for that particular subject. Thus, raw marks (for single session test papers) or normalized marks (for multi-session test papers) will be used for computing the GATE Score, based on the qualifying marks.

Calculation of Normalized Marks for Multi-Session Papers

In GATE 2027 examination, some test papers may be conducted in multi-sessions. For such test papers, a suitable normalisation is applied to take into account any variation in the difficulty levels of the question papers across sessions. The normalization is done based on the fundamental assumption that "in all multi-session GATE test papers, the distribution of abilities of candidates is the same across all the sessions". This assumption is justified since the number of candidates appearing in multisession test papers in GATE 2027 will be large and the procedure for allocation of session to candidates is random. Further, it is also ensured that for the same multi-session test paper, the number of candidates allotted in each session is of the same order of magnitude.

Based on the above, and considering various normalization methods, the committee arrived at the following formula for calculating the normalized marks for the multi-session papers.

The normalised marks of the j^{th} candidate in the i^{th} session, denoted by $\widehat{M}_{ij}$ are computed as

$$\widehat{M}_{ij} = \frac{\overline{M}_t^g - M_q^g}{\overline{M}_{ti} - M_{iq}}\left(M_{ij} - M_{iq}\right) + M_q^g$$

where,

M_{ij}, is the actual marks obtained by the j^{th} candidate in the i^{th} session;

$\overline{M}_t^g$ is the average marks of the top 0.1% of the candidates considering all sessions;

M_q^g is the sum of mean and standard deviation marks of the candidates in the paper considering all sessions;

$\overline{M}_{ti}$ is the average marks of the top 0.1% of the candidates in the i^{th} session; and

$\overline{M}_{iq}$ is the sum of the mean marks and standard deviation marks of the i^{th} session.

Calculation of GATE Score for All Test Papers

For all papers for which there is only one session, actual marks obtained by the candidates will be used for calculating the GATE 2027 Score. For papers in multi-sessions, normalized marks will be calculated corresponding to the raw marks obtained by a candidate and the GATE 2027 Score will be calculated based on the normalized marks.

The GATE 2026 score will be computed using the formula given below.

$$\text{GATE Score} = S_q + (S_t - S_q)\frac{M - M_q}{\overline{M_t} - M_q}$$

where

M is the marks obtained by the candidate (actual marks for single session papers and normalized marks for multi-session papers);

M_q is the qualifying marks for the General Category candidate in the paper;

$\overline{M_t}$ is the mean of marks of top 0.1% or top 10 (whichever is larger) of the candidates who appeared in the paper (in case of multi-session papers including all sessions);

$S_q = 350$ is the score assigned to M_q ; and

$S_t = 900$ is the score assigned to $\overline{M_t}$

In GATE 2026, the qualifying marks (M_q) for the general category candidate in each subject will be

Cut-off marks for GENERAL category$= \max(25, \min(40, \ \mu + \sigma))$

where μ denotes the mean of marks of all candidates and a denotes the standard deviation of positive marks of all the candidates who appeared for the test paper. The cut-off marks for the other categories may be calculated as per the standard ratio:

Cut-off for OBC-NCL/EWS categories = (9/10) × GENERAL cut-off

Cut-off for SC/ST/PWD categories = (2/3) × GENERAL cut-off

GATE Year Wise Organising Institute

Organizing Year	Organising Institute
2027	IIT Madras
2026	IIT Guwahati
2025	IIT Roorkee
2024	IISc Bengaluru

CUT-OFF

YEAR	General	OBC	SC/ST
2026	26.4	23.7	17.5
2025	29	26.1	19.3
2024	37.1	33.3	24.7

CONTENT

DATA SCIENCE AND ARTIFICIAL INTELLIGENCE

1 GENERAL APTITUDE 1

1.1: *Verbal Aptitude 1*

1.2: *Quantitative Aptitude 9*

1.3: *Analytical Aptitude 29*

1.4: *Spatial Aptitude 38*

2 PROBABILITY & STATISTICS 43

2.1: *Fundamentals of Probability 44*

2.2: *Random Variables and Probability Distributions 59*

2.3: *Descriptive Statistics 93*

3 Linear Algebra 97

3.1: *Matrices & Their Properties 98*

3.2: *Vector Spaces 109*

3.3: *System of Equations 117*

3.4: *Eigenvalues & Eigenvectors 123*

4 Calculus and Optimization 131

4.1: *Limit and Continuity 132*

4.2: *Differentiation, Taylor's Theorem, Maxima Minima 138*

5 Programming, Data Structures and Algorithms 153

5.1: *Programming in Python 154*

5.2: *Stack, Queue and Linked List 166*

5.3: *Trees 176*

5.4: *Hash table 184*

5.5: *Searching Algorithm 190*

5.6: *Basic Sorting Algorithm 193*

5.7: *Graph 200*

6 Database Management and Warehousing 214

6.1: *ER Model & Relational Model 215*

6.2: *Relational Algebra and Tuple Calculus ... 219*

6.3: *SQL 228*

6.4: *Normal Form 247*

6.5: *Transaction & Concurrency Controller ... 253*

6.6: *Data Warehouse Modelling 257*

6.7: *Data Transformation & Comparison 260*

7 Machine Learning 264

7.1: *Supervised Learning: Basics and Regression 265*

7.2: *Supervised Learning: Classification Algorithms 272*

7.3: *Model Evaluation & Regularization 283*

7.4: *Supervised Learning: Classification Algorithms 287*

7.5: *Unsupervised Learning: Clustering Algorithms 292*

7.6: *Unsupervised Learning: Dimensionality Reduction 299*

8 Artificial Intelligence 305

8.1: *Search Algorithms 306*

8.2: *Logic and Representation 316*

8.3: *Reasoning under Uncertainty 323*

MOCK TEST

1. *Mock Test 328*
2. *Mock Test 345*
3. *Mock Test 363*
4. *Mock Test 381*
5. *Mock Test 399*

NUMBER OF QUESTIONS ASKED IN PREVOUS EXAM (2024 to 2026)

DATA SCIENCE AND ARTIFICIAL INTELLIGENCE

1	**General Aptitude**	**2024**	**2025**	**2026**
	Unit 1.1: Verbal Aptitude	2	3	3
	Unit 1.2: Quantitative Aptitude	6	4	4
	Unit 1.3: Analytical Aptitude	1	3	2
	Unit 1.4: Spatial Aptitude	1	-	1
2	Probability & Statistics	**2024**	**2025**	**2026**
	Unit 2.1 Fundamentals of Probability	2	3	3
	Unit 2.2 Random Variables and Probability Distributions	6	11	9
	Unit 2.3 Descriptive Statistics	1	-	1
3	**Linear Algebra**	**2024**	**2025**	**2026**
	Unit 3.1: Matrices & Their Properties	1	5	3
	Unit 3.2: Vector Spaces	2	1	1
	Unit 3.3: System of Equations	1	1	-
	Unit 3.4: Eigenvalues & Eigenvectors	2	1	1
4	**Calculus and Optimization**	**2024**	**2025**	**2026**
	Unit 4.1: Limit and Continuity	1	1	3
	Unit 4.2: Differentiation, Taylor's Theorem, Maxima Minima	4	5	1
5	**Programming, Data Structures and Algorithms**	**2024**	**2025**	**2026**
	Unit 5.1: Programming in Python	3	3	4
	Unit 5.2: Stack, Queue and Linked List	2	1	-
	Unit 5.3: Trees	2	-	4
	Unit 5.4: Hash table	1	1	-
	Unit 5.5: Searching Algorithm	1	1	-
	Unit 5.6: Basic Sorting Algorithm	2	1	3
	Unit 5.7: Graph	3	2	1
6	**Database Management and Warehousing**	**2024**	**2025**	**2026**
	Unit 6.1: ER Model & Relational Model	-	-	1
	Unit 6.2: Relational Algebra and Tuple Calculus	1	2	2
	Unit 6.3: SQL	2	4	3
	Unit 6.4: Normal Form	1	1	1
	Unit 6.5: Transaction & Concurrency Controller	-	-	-
	Unit 6.6: Data Warehouse Modelling	-	-	-
	Unit 6.7: Data Transformation & Comparison	-	-	-
7	**Machine Learning**	**2024**	**2025**	**2026**
	Unit 7.1: Supervised Learning: Basics and Regression	3	2	2
	Unit 7.2: Supervised Learning: Classification Algorithms	3	3	3
	Unit 7.3: Model Evaluation & Regularization	1	-	-
	Unit 7.4: Supervised Learning: Classification Algorithms	1	2	-
	Unit 7.5: Unsupervised Learning: Clustering Algorithms	2	1	3
	Unit 7.6: Unsupervised Learning: Dimensionality Reduction	3	-	1
8	**Artificial Intelligence**	**2024**	**2025**	**2026**
	Unit 8.1: Search Algorithms	2	2	1
	Unit 8.2: Logic and Representation	2	1	4
	Unit 8.3: Reasoning under Uncertainty	-	-	-

Chapter 5

Programming, Data Structures and Algorithms

GATE DATA SCIENCE & ARTIFICIAL INTELLIGENCE

Unit 5.1: Programming in Python

[PREVIOUS YEAR EXAM QUESTIONS]

Multiple Choice Questions (MCQ)

[2024: 1 Mark]

1. Consider the following Python code:

```
def count(child_dict, i):
    if i not in child_dict.keys():
        return 1
    ans = 1
    for j in child_dict[i]:
        ans += count(child_dict, j)
    return ans
child_dict = dict()
child_dict[0] = [1, 2]
child_dict[1] = [3, 4, 5]
child_dict[2] = [6, 7, 8]
print(count(child_dict, 0))
```

Which ONE of the following is the output of this code?

(a) 6 (b) 1

(c) 8 (d) 9

[2024: 2 Marks]

2. Consider the function computeS(X) whose pseudocode is given below:

```
ComputeS(X)
S[1] ← 1
for i ← 2 to length(X)
 S[i] ← 1
 if X[i − 1] ≤ X[i]
     S[i] ← S[i] + S[i − 1]
  end if
end for
return S
```

Which ONE of the following values is returned by the function computeS(X) for $X = [6, 3, 5, 4, 10]$?

(a) [1, 1, 2, 3, 4] (b) [1, 1, 2, 3, 3]

(c) [1, 1, 2, 1, 2] (d) [1, 1, 2, 1, 5]

[2024: 2 Marks]

3. Consider the following Python function:

```
def fun(D, s1, s2):
  if s1 < s2:
    D[s1], D[s2] = D[s2], D[s1]
   fun(D, s1+1, s2-1)
```

What does this Python function fun () do? Select the ONE appropriate option below.

(a) Reverses the elements of the list D between indices s1 and s2.

(b) Sorts the elements of the list D in ascending order.

(c) Swaps the first and last elements of the list D.

(d) Flips the bits of the elements in the list D.

[2025: 1 Mark]

4. Consider the following Python declarations of two lists:

```
A = [1, 2, 3]
B = [4, 5, 6]
```

Which one of the following statements results in $A = [1, 2, 3, 4, 5, 6]$?

(a) A.extend(B) (b) A.append(B)

(c) A.update(B) (d) A.insert(B)

[2025: 2 Marks]

5. Consider the following Python code snippet:

```
A = {"this", "that"}
B = {"that", "other"}
C = {"other", "this"}
while "other" in C:
  if "this" in A:
    A, B, C = C | A, A | B, B | C
  if "that" in B:
    A, B, C = C | A, B | C, A | B | C
```

When the above program is executed, at the end, which of the following sets contains "this"?

(a) Only A (b) Only B

(c) Only C (d) A and C

[2026: 1 Mark]

6. Consider the given Python program.

```
def append _ to _ 1st (va1, 1st=[]):
  1st.append(va1)
  return 1st
print(append_to_1st(1))
print(append_to_1st(2))
print(append_to_1st(3, []))
```

Which of the following is the correct output of this program?

(a) [1]
[2]
[3]

(b) [1]
[1, 2]
[3]

(c) [1]
[2]

[1, 2, 3]
(d) [1]
[1, 2]
[1, 3]

[2026: 2 Marks]

7. A recursive function in Python is given.

```
def mystery(n):
  if n <= 0:
    return 1
  else:
    return mystery(n-1) + mystery(n-2)
```

Now, consider the following function call:
Mystery (4)
Assume that a typical runtime stack is used to manage function calls. Each function call is pushed onto the stack and removed only after it finishes execution.
Which of the following options denotes the total number of function calls (i.e., the total number of stack activations), including the initial call, to compute mystery (4)?

(a) 5 (b) 9
(c) 15 (d) 17

Multiple Select Questions (MSQ)

[2026: 2 Marks]

1. Consider the given Python program.

```
def outer():
    x = []
    def inner(val):
      x.append(val)
      return x
    return inner
f1 = outer()
f2 = outer()
print(f1(10)) # Line P
print(f1(20)) # Line Q
print(f2(30)) # Line R
print(f1(40)) # Line S
```

Which of the following options is/are correct?
(a) f1 and f2 share the same list x
(b) Output of Line Q is [10, 20]
(c) Output of Line R is [10, 20, 30]
(d) Output of Line S is [10, 20, 40]

Numerical Answer Type (NAT)

[2025: 2 Marks]

1. Consider the following Python code snippet:

```
def f(a, b):
    if (a == 0):
      return b
    if (a % 2 == 1):
      return 2 * f((a - 1) / 2, b)
    return b + f(a - 1, b)
print(f(15, 10))
```

The value printed by the code snippet is _____ (Answer in integer)

[2026: 2 Marks]

2. Consider the given Python program.

```
def fun(L,  i=0):
    if i >= len(L)-1:
     return 0
  if L [i]  > L [i+1] :
      L[i+1], L[i] = L[i], L[i+1]
return 1 + fun(L, i+1)
else:
    return fun(L, i+1)
data = [5, 3, 4, 1, 2]
count = 0
for _ in range( len (data)):
    count += fun(data)
print(count)
```

The output of the program is _____ . (Answer in integer)

ANSWER KEY				
Multiple Choice Questions (MCQ)				
1	**2**	**3**	**4**	**5**
(d)	(c)	(a)	(a)	(b)
6	**7**			
(b)	(c)			
Multiple Select Questions (MSQ)				
1				
(b, d)				
Numerical Answer Type (NAT)				
1		**2**		
(160 to 160)		(8 to 8)		

[PRACTICE QUESTIONS]

Multiple Choice Questions (MCQ)

1. Recursive Dictionary Tree Count

```
def dfs_count(tree, node):
    if node not in tree:
        return 1
return 1 + sum(dfs_count(tree, child) for child in tree[node])
tree = {
    'A': ['B', 'c'],
    'B': ['D', 'E'],
    'C': ['F']
}
print(dfs_count (tree, 'A'))
```

What is the output of the code?

(a) 3 (b) 6
(c) 7 (d) 8

2. Consider below code

```
def depth (tree, node):
    if node not in tree or len(tree[node]) = = 0:
        return 1
    return 1 + max(depth (tree, child) for child in tree[node])
tree = {
    0: [1, 2],
    1: [3],
    3: [4]
}
print(depth(tree, 0))
```

What is the output of this code?

(a) 2 (b) 3
(c) 4 (d) 5

3. Consider the following pseudocode

```
TransformArray(A)
    B[1] ← A[1]
    for i ← 2 to length(A)
        if A[i] > A[i - 1]
            B[i] ← B[i - 1] + 1
        else
            B[i] ← 1
    return B
```

Input: A = [1, 2, 2, 3, 4, 1, 2]

What is the output of TransformArray(A) ?

(a) [1, 2, 1, 2, 3, 1, 2] (b) [1, 2, 1, 2, 3, 1, 3]
(c) [1, 2, 2, 3, 4, 1, 2] (d) [1, 2, 2, 3, 4, 5, 6]

4. Consider the function AssignRank(X) whose pseudocode is given below:

```
AssignRank(X)
    R[1] ← 1
    for i ← 2 to length(X)
        if X[i] > X[i - 1]
            R[i] ← R[i - 1] + 1
        else if X[i] ==X[i-1]
            R[i] ← R[i - 1]
        else
            R[i] ← 1
    return R
```

Input: X = [3, 4, 4, 2, 5]

What is the output of AssignRank (X)?

(a) [1, 2, 3, 1, 2] (b) [1, 2, 2, 1, 2]
(c) [1, 1, 2, 1, 2] (d) [1, 2, 3, 2, 3]

5. What will be the value of A after the following operations?

A = [1, 2, 3]
A.insert(1, [10, 20])

(a) [1, 2, 3, 10, 20] (b) [1, [10, 20], 2, 3]
(c) [1, 10, 20, 2, 3] (d) [1, 2, [10, 20], 3]

6. What will the following code output in Python 3?

```
a = [1, 2, 3]
    b = a
    a += [4, 5]
    print(b)
```

(a) [1, 2, 3] (b) [1, 2, 3, 4, 5]
(c) [4, 5] (d) Error

7. Which of the following best describes Python's list slicing a[start: stop:step]?

(a) Returns every element from index start to stop (inclusive)
(b) Returns elements starting at start, skipping step elements, up to (but not including) stop
(c) Returns every element from start to stop including stop, skipping step elements
(d) Returns elements between start and stop with step of 1, always

8. What is the output of the following?

```
a = {1, 2, 3}
b = {3, 4, 5}
print(a & b)
```

(a) {1, 2, 3, 4, 5} (b) {3}
(c) {1, 2} (d) {}

9. What will be the result of this code snippet?

```
X = [i**2 for i in range(5) if i % 2 == 0]
print(x)
```

(a) [0, 4, 16] (b) [1, 9, 25]
(c) [0, 1, 4] (d) [0, 2, 4]

10. What will be the output of the following code?

```
def foo(1st):
return [x for x in 1st if all(x % y != 0 for y in range(2, x))]
print (foo([2, 3, 4, 5, 6, 7, 8]))
```

(a) [2, 3, 5, 7] (b) [2, 3, 4, 5, 6, 7, 8]
(c) [2, 3, 5, 7, 8] (d) [3, 5, 7]

11. What will be the output of this code involving sets?

```
a = {1, 2, 3}
b = {3, 4, 5}
a.symmetric_difference_update(b)
print(a)
```

(a) {1, 2, 3, 4, 5} (b) {1, 2, 4, 5}
(c) {3, 4, 5} (d) {1, 2}

12. Which of the following statements will result in a KeyError?

{'a': 1, 'b': 2, 'c': 3}
(a) d.get('d')
(b) d['d']
(c) 'd' in d
(d) d.setdefault('d', 4)

13. What is the result of the following list slicing in Python?
a = = [10, 20, 30, 40, 50]
print(a[::2])
(a) [10, 30, 50] (b) [10, 20, 30]
(c) [30, 50] (d) [20, 40]

Multiple Select Questions (MSQ)

1. Consider the following code
A = [1, 2, 3]
B = A
B.append(4)
Which of the following are correct?
(a) A and B refer to the same list in memory
(b) A == B is True
(c) A is B is True
(d) A will be [1, 2, 3] after the append

2. a, *b, c = [1, 2, 3, 4, 5]
Which of the following are correct?
(a) a = 1, c = 5
(b) b = [2, 3, 4]
(c) This syntax is invalid in Python
(d) The starred variable b collects all values between the first and last

3. Consider the code below
A = {1, 2, 3}
B = {3, 4}
Which of the following modify set A?
(a) A.update(B)
(b) A.union(B)
(c) A. symmetric_difference_update(B)
(d) A = A.intersection(B)

4. Consider the code below
d = {'x': 1, 'y': 2}
Which of the following will safely check for a key or return a value?
(a) 'z' in d
(b) d['z']
(c) d.get('z') is None
(d) d.setdefault('z', 0) == 0

5. Consider the code below
x = 5
y = 2
Which expressions will return True?
(a) x & y == 0 (b) x | y > x
(c) x ^ y < x + y (d) not (x and y)

6. Consider the code below
s1 = "gate"
s2 = "gate"
s3 = " ".join(["g", "a", "t", "e"])
Which of the following are True?
(a) s1 == s2 (b) s1 is s2
(c) s1 == s3 (d) s1 is s3

7. Consider the code below
x = [10, 20]
y = x
y.append(30)
Which are True after execution?
(a) x is y (b) x == y
(c) x = [10, 20] (d) len(x) == 3

Numerical Answer Type Questions (NAT)

1. Consider the following Python code snippet:
```
def power (x, n):
if n == 0:
return 1
if n % 2 == 0:
return power(x * x, n // 2)
return x * power(x * x, (n - 1) // 2)
print(power (3, 5))
```
The value printed by the code snippet is _____ (Answer in integer)

2. Consider the following Python code snippet:
```
def compute(a, b):
if a == 0:
return 0
if a % 2 == 0:
return compute(a // 2, b * 2)
return b + compute(a-1, b)
print(compute(13, 5))
```
The value printed by the code snippet is _____ (Answer in integer)

3. Consider the following Python code snippet:
```
count = 0
def recursive(n):
global count
if n <= 1:
return 1
count += 1
return recursive(n - 1) + recursive(n - 2)
recursive(5)
print(count)
```
The value printed by the code snippet is _____ (Answer in integer)

4. How many elements will be present in the resulting set after executing the following code?
python
a = {10, 20, 30, 40}
b = {30, 50, 60}

c = a.union(b).intersection({20, 30, 50})
The value printed by the code snippet is _____ (Answer in integer)

5. **What is the ASCII value of the last character in this output?**
S ="GATEExam2025"
Output =s[::-2]
print(output)
Enter the ASCII value of the last character of the string printed.

6. **Consider code below**
a = [1, 2, 2, 3, 4, 4, 5]
b = set(a)
c = list(b)
c.sort()
print(c[2])
What value is printed?

7. **Consider code below**
A = {i for i in range(1, 30) if i % 2 == 0}
B = {i for i in range(10, 50) if i % 3 == 0}
C = A & B
Enter the number of elements in set C.

8. **Consider the code below**
A = {x for x in range(1, 51) if x % 2 ==0}
B = {x for x in range(1, 51) if x % 5 == 0}
C = A.intersection(B)
Enter the sum of all elements in C.

9. **Consider the code below**
count = 0
n = 128
while n > 1:
n = n // 2
count += 1
Enter the final value of count.

10. **Consider the code below**
nums = [x for x in range(1, 20) if x % 2 == 0 and x % 3 == 0]
Enter the sum of all elements in nums.

ANSWER KEY

Multiple Choice Question (MCQ)

1	2	3	4	5	6	7	8	9	10
(a)	(c)	(a)	(b)	(b)	(b)	(b)	(b)	(a)	(a)
11	**12**	**13**							
(b)	(b)	(a)							

Multiple Select Question (MSQ)

1	2	3	4	5
(a, b, c)	(a, b, d)	(a, c)	(a, c, d)	(a, b)
6	**7**			
(a, b, c)	(a, b, d)			

Numerical Answer Type Question (NAT)

1	2	3	4	5
(243)	(65)	(7)	(3)	(69)
6	**7**	**8**	**9**	**10**
(3)	(3)	(150)	(7)	(36)

EXPLANATIONS

[PREVIOUS YEAR EXAM QUESTIONS]

Multiple Choice Questions (MCQ)

1. **Explanation: (d)**
- The function counts the total number of nodes in the tree rooted at node i.
- If a node has no children (not in child_dict), it counts as 1.
- For node 0, it has children 1 and 2.
- Node 1 has 3 children (3, 4, 5), all leaves → counts as 4 nodes total (1 + 3).
- Node 2 has 3 children (6, 7, 8), all leaves → counts as 4 nodes total.
- So total nodes under 0 = 1 (itself) + 4 (subtree 1) + 4 (subtree 2) = 9.

Hence, the output is 9.

2. **Explanation: (c)**
Let's trace the function computeS(X) step-by-step for the input:
$X = [6,3,5,4,10]$
Initially:
- $S[1] \leftarrow 1$
- For $i = 2$ to 5:
- $S[i] \leftarrow 1$
- If $X[i-1] \leq X[i]$, then $S[i] \leftarrow S[i] + S[i-1]$

Now,
- **i = 1:**
$S[1] = 1$ (given)
- **i = 2:**
$S[2] = 1$
Check if $X[1] \leq X[2] \Rightarrow 6 \leq 3$? No → no addition
So, $S[2] = 1$
- **i = 3:**
$S[3] = 1$
Check if $X[2] \leq X[3] \Rightarrow 3 \leq 5$? Yes →
$S[3] = S[3] + S[2] = 1 + 1 = 2$
- **i = 4:**
$S[4] = 1$
Check if $X[3] \leq X[4] \Rightarrow 5 \leq 4$? No → no addition
So, $S[4] = 1$
- **i = 5:**
$S[5] = 1$
Check if $X[4] \leq X[5] \Rightarrow 4 \leq 10$? Yes →

$S[5] = S[5] + S[4] = 1 + 1 = 2$
Final $S = [1,1,2,1,2]$
So, Correct option is c.

3. Explanation: (a)
This function:
1. Swaps the elements at positions s1 and s2.
2. Then calls itself with s1+1 and s2-1.
3. This continues until s1 is no longer less than s2.

So, it reverses the part of the list from index s1 to s2.
Example:
If D = [10, 20, 30, 40, 50]
and we call fun (D, 1, 3)
Then the result will be: [10, 40, 30, 20, 50]
which means the sublist [20, 30, 40] was reversed to [40, 30, 20]
Correct Answer: (a) Reverses the elements of the list D between indices s1 and s2.

4. Explanation: (a)
Use extend() when you want to merge two lists **element by element**.
So, Correct option is a.

5. Explanation: (b)
Initial values:
- A = {"this", "that"}
- B = {"that", "other"}
- C = {"other", "this"}

Loop condition:
While "other" in C: → True initially.
First if "this" in A: → True
So update:
python
A, B, C = C | A, A | B, B | C
Using current values:
- C | A = {"this", "other"} | {"this", "that"} = {"this", "that", "other"} → A
- A | B = {"this", "that"} | {"that", "other"} = {"this", "that", "other"} → B
- B | C = {"that", "other"} | {"this", "other"} = {"this", "that", "other"} → C

All sets now: {"this", "that", "other"}
Next if "that" in B: → True
So update:
python
A, B, C = C | A, B | C, A | B
Again, all values are already the same
sets remain unchanged.
Copy Edit
Copy Edit
Final state:
- A = {"this", "that", "other"}
- B = {"this", "that", "other"}
- C = {"this", "that", "other"}

So "this" is in A, B, and C
But based on right-hand side evaluation (before assignment), in the first update, only B gains "this" (initially it didn't have it).
So just after one iteration, only B newly gets "this".
Final Answer: (b) Only B.

6. Explanation: (b)
In Python, **default arguments are evaluated only once** at function definition.
The default list [] is the **same list object** reused across calls that don't provide their own list.
Step-by-step:
1. append_to_lst(1) → uses default list → becomes [1] → prints [1]
2. append_to_lst(2) → uses **same** default list → now [1, 2] → prints [1, 2]
3. append_to_lst(3, []) → explicit new list [] → becomes [3] → prints [3]

Output:
[1]
[1, 2]
[3]
So, Correct option is B.

7. Explanation: (c)
The function mystery(n) is like a Fibonacci recursion but with base case returning 1 for n ≤ 0.
For mystery(4):
Call tree:
mystery(4)
├── mystery(3)
│ ├── mystery(2)
│ │ ├── mystery(1)
│ │ │ ├── mystery(0)
│ │ │ └── mystery(-1)
│ │ └── mystery(0)
│ └── mystery(1)
│ ├── mystery(0)
│ └── mystery(-1)
└── mystery(2)
├── mystery(1)
│ ├── mystery(0)
│ └── mystery(-1)
└── mystery(0)
Counting all nodes:
Total nodes = 15.

Multiple Select Questions (MSQ)

1. Explanation: (b, d)
outer() creates a new list x each time it's called and returns inner() which remembers that specific list.
f1 = outer() → f1 gets its own list x = []

f2 = outer() → f2 gets a **different** list x = []
Execution:
f1(10) → appends 10 → [10]
f1(20) → appends 20 → [10, 20] → **Option (B) correct**
f2(30) → appends 30 to its own empty list → [30] → **Option (C) wrong**
f1(40) → appends 40 → [10, 20, 40] → **Option (D) correct**
Option (A) is wrong because f1 and f2 have separate lists.
So, Correct options are B, D.

Numerical Answer Type (NAT)

1. Explanation: (160 to 160)
For f(15, 10):
- 15 is odd → f(15, 10) = 2 * f(7, 10)
- 7 is odd → 2 * f(3, 10)
- 3 is odd → 2 * f(1, 10)
- 1 is odd → 2 * f(0, 10)
- f(0, 10) returns **10**

Now backtrack:
- f(1, 10) = 2 * 10 = 20
- f(3, 10) = 2 * 20 = 40
- f(7, 10) = 2 * 40 = 80
- f(15, 10) = 2 * 80 = 160

So, Correct answer will be 160.

2. Explanation: (8)
Initial: data = [5, 3, 4, 1, 2]
Iteration 1: fun([5,3,4,1,2])
- i=0: 5>3 → swap → [3,5,4,1,2] → return 1 + fun(L,1)
- i=1: 5>4 → swap → [3,4,5,1,2] → return 1 + fun(L,2)
- i=2: 5>1 → swap → [3,4,1,5,2] → return 1 + fun(L,3)
- i=3: 5>2 → swap → [3,4,1,2,5] → return 1 + fun(L,4)
- i=4: i>=4 (len=5) → return 0

Backtrack: swaps count = 1+1+1+1 = **4**
count += 4 → count = 4
Now data = [3,4,1,2,5]
Iteration 2: fun([3,4,1,2,5])
- i=0: 3>4? No → fun(L,1)
- i=1: 4>1 → swap → [3,1,4,2,5] → return 1 + fun(L,2)
- i=2: 4>2 → swap → [3,1,2,4,5] → return 1 + fun(L,3)
- i=3: 4>5? No → fun(L,4)
- i=4: base → 0

Backtrack: swaps = 0 + 1 + 1 + 0 = **2**
count += 2 → count = 6
Now data = [3,1,2,4,5]
Iteration 3: fun([3,1,2,4,5])
- i=0: 3>1 → swap → [1,3,2,4,5] → return 1 + fun(L,1)
- i=1: 3>2 → swap → [1,2,3,4,5] → return 1 + fun(L,2)
- i=2: 3>4? No → fun(L,3)
- i=3: 4>5? No → fun(L,4)
- i=4: base → 0

Swaps = 1 + 1 + 0 + 0 = **2**
count += 2 → count = 8
Now data = [1,2,3,4,5]
Iteration 4: fun([1,2,3,4,5])
No swaps at any i → total = **0**
count += 0 → count = 8
Iteration 5: same, no swaps → **0**
count += 0 → count = 8
So, Correct answer is 8.

[PRACTICE QUESTIONS]

Multiple Choice Questions (MCQ)

1. Explanation: (a)
We start with the call:
Python
dfs_count(tree, 'A')
tree['A']=['B', 'C']
So,
python
dfs_count(tree, 'A') = 1 + dfs_count(tree, 'B') + dfs_count (tree, 'C')
Now evaluate dfs_count(tree, 'B'):
- **tree['B'] = ['D', 'E']**

python
dfs_count(tree, 'B') = 1 + dfs_count(tree, 'D') + dfs_count(tree, 'E')
- 'D' is not a key in the dictionary, so dfs_count(tree, 'D')
- 'E' is not a key either, so dfs_count(tree, 'E') = 1

So,
python
dfs_count(tree, 'B') = 1 + 1 + 1 = 3
Now evaluate dfs_count (tree, 'C'):
- tree['C'] = ['F']

dfs_count(tree, 'C') = 1 + dfs_count (tree, 'F')`
- 'F' is not a key in the dictionary, so dfs_count(tree, 'F') = 1

So,
python
dfs_count (tree, 'C') = 1 + 1 = 2
Finally,
python
dfs_count(tree, 'A') = 1 + 3 + 2 = 6
Final Answer: 6
Option (a) is correct.

2. Explanation: (c)
We begin with the call:
python
depth(tree, 0)
tree[0] = [1, 2]
So:
python
depth(0) = 1 + max(depth(1), depth(2))

First, compute depth(1):
tree[1] = [3]
depth(1) = 1 + max(depth(3))
tree[3] = [4]
python
depth(3) = 1 + max(depth(4))
4 is not in the dictionary, so:
python
depth(4) = 1
Therefore:
depth(3) = 1 + 1 = 2
depth(1) = 1+2 = 3
Now compute depth(2):
2 is not in the dictionary, so:
python
depth(2) = 1
Now go back to the original call:
python
depth(0) = 1 + max(3, 1) = 1 + 3 = 4
Final Answer: 4
Option (c) is correct.

3. Explanation: (a)
Let us initialize:
- B[1] = A[1] = 1

Now loop from i = 2 to 7 (length of A = 7)
- i = 2, A[2] = 2, A[1] = 1 → A[2] > A[1] → B[2] = B[1] + 1 = 1 + 1 = 2
- i = 3, A[3] = 2, A[2] = 2 → A[3] is not greater →B[3] = 1
- i = 4, A[4] = 3, A[3] = 2 → A[4] > A[3] → B[4] = B[3] + 1 = 1 + 1 = 2
- i = 5, A[5] = 4, A[4] = 3 → A[5] > A[4] → B[5] = B[4] + 1 = 2 + 1 = 3
- i = 6, A[6] = 1, A[5] = 4 → A[6] < A[5] → B[6] = 1

Final B:
csharp
[1, 2, 1, 2, 3, 1, 2]
Correct Answer: (a) [1, 2, 1, 2, 3, 1, 2]

4. Explanation: (b)
Step-by-step Execution:
Initialize:
- R[1] = 1

Loop from i = 2 to 5:
- i = 2, X[2] = 4, X[1] = 3 → 4 > 3 → R[2] = R[1] + 1 = 1 + 1 = 2
- i = 3, X[3] = 4, X[2] = 4 → equal → R[3] = R[2] = 2
  i = 4, X[4] = 2, X[3] = 4 → 2 <4 → R[4] = 1
- i = 5, X[5] = 5, X[4] = 2 → 5 > 2 → R[5] = R[4] + 1 = 1 + 1 = 2

Final R:
csharp
[1, 2, 2, 1, 2]
Correct Answer: (b) [1, 2, 2, 1, 2]

5. Explanation: (b)
.insert() inserts a single item at a specified index.
The method A.insert(index, element) inserts the entire object element at the specified index.
- A = [1, 2, 3]
- A.insert(1, [10, 20]) inserts the list [10, 20] at index 1

So the updated list A becomes:
[1, [10, 20], 2, 3]
Final Answer: (b) [1, [10, 20], 2, 3]

6. Explanation: (b)
- a = [1, 2, 3] creates a list a.
- b = a means b references the same list object as a
- a += [4, 5] is in-place extension of the list a (like a.extend([4, 5])).
- Since brefers to the same object, changes in a will reflect in b.

After a += [4, 5]:
- a becomes [1, 2, 3, 4, 5]
- bis also [1, 2, 3, 4, 5]

Final Output:
python
[1, 2, 3, 4, 5]
Correct Option: (b) [1, 2, 3, 4, 5]

7. Explanation: (b)
a = [0, 1, 2, 3, 4, 5, 6]
a[1:6:2] → [1, 3, 5]
This starts at index 1, goes up to but does not include index 6, and selects every second element.
Now evaluate the options:
(a) Returns every element from index start to stop (inclusive)
Incorrect stop is exclusive
(b) Returns elements starting at start, skipping step elements, up to (but not including) stop
Correct - this matches Python slicing behavior
(c) Returns every element from start to stop including stop, skipping step elements
Incorrect — stop is not included
(d) Returns elements between start and stop with step of 1, always step can be any integer
Incorrect
Correct answer: (b)

8. Explanation: (b)
- a & b performs set intersection.
- It returns a set of elements that are common to both a and b.

a = {1, 2, 3}
b = {3, 4, 5}
The only common element is 3.
So,
python
a & b = {3}
Correct answer: (b) {3}

9. Explanation: (a)

- range (5) generates: [0, 1, 2, 3, 4]
- The condition if i % 2 == 0 filters even numbers: [0, 2, 4]
- For each of these, compute i**2:
- 0**2 = 0
- 2**2 = 4
- 4**2 = 16

So, the list x becomes: [0, 4, 16]
Correct answer: (a) [0, 4, 16]

10. Explanation: (a)

This function returns a list of numbers from 1st such that each number x is not divisible by any number y in the range 2 to x-1. That means it's checking for prime numbers.
Let's check each number from the list [2, 3, 4, 5, 6, 7, 8]:

- 2: range (2,2) is empty all([]) is True → include
- 3: range (2,3) check if 3 % 20 → True → include
- 4: check 4 % 2 = 0 False exclude
- 5: 5 % 2 * 0, 5 % 3 ≠ 0, 5 % 4 ≠ 0 → True → include
- 6:6% 2 = 0 → False → exclude
- 7:7% 2≠0,7% 3≠0,7 % 4≠0,7 % 5≠0, 7 % 6 ≠ 0 → include
- 8: 8% 2 = 0 → False → exclude

Output: [2, 3, 5, 7]
So correct answer is (a).

11. Explanation: (b)

symmetric_difference_update (b) modifies a to contain elements that are in either a or b but not both.

- a = {1, 2, 3}
- b = {3, 4, 5}

Common element = {3}
Symmetric difference = elements in either set but not both = {1, 2, 4, 5}
So a becomes {1, 2, 4, 5}.
Final Answer: (b) {1, 2, 4, 5}

12. Explanation: (b)

We evaluate each option:
(a) d.get('d')

- get() returns None if the key is not found.
- No KeyError.

(b) d['d']

- This tries to access a non-existent key directly.
- Raises KeyError.

(c) 'd' in d

- Just checks existence of the key.
- Returns False, no error.
- No KeyError.

(d) d.setdefault('d', 4')

- Adds 'd': 4 if 'd' not present.
- No KeyError.

Final Answer: (b) d['d']

13. Explanation: (a)

The slicing syntax a[::2] means:

- Start from the beginning (default start)
- Go till the end (default stop)
- Step = 2 → pick every second element

List indices:
a[0] = 10
a[1] = 20
a[2] = 30
a[3] = 40
a[4] = 50
Now picking every 2nd element from index 0:
[a[0], a[2], a[4] =[10, 30, 50]]
Final Answer: (a) [10, 30, 50]

Multiple Select Questions (MSQ)

1. Explanation: (a, b, c)

Now evaluate each statement:
(a) A and B refer to the same list in memory
B = A assigns the same reference.
True
(b) A == B is True
Both point to the same list → contents are equal.
True
(c) A is B is True
They are the same object (same memory address).
True
(d) A will be [1, 2, 3] after the append
Since both A and B refer to the same object, the append affects both.
A becomes [1, 2, 3, 4]
False
Correct Answers: (a), (b), (c)

2. Explanation: (a, b, d)

Now evaluate each option:
(a) a = 1, c = 5
→ Correct
True
(b) b = [2, 3, 4]
→ Correct
True
(c) This syntax is invalid in Python

→ Incorrect; this is valid in Python 3.x
False

(d) The starred variable b collects all values between the first and last
→ Exactly what it does
True
Correct Answers: (a), (b), (d)

3. Explanation: (a, c)

(a) A.update(B)
This modifies set A in-place by adding elements from B.
After execution, A becomes {1, 2, 3, 4}.
This modifies A.

(b) A.union(B)
This returns a new set that is the union of A and B, but does not change A itself.
A remains unchanged.
This does not modify A.

(c) A. symmetric_difference_update(B)
This modifies A in-place to hold the symmetric difference (elements in either A or B but not both).
After execution, A becomes {1, 2, 4}.
This modifies A.

(d) A = A.intersection (B)
This creates a new set from the intersection of A and B, and reassigns it to variable A.
However, the original set object is not modified in-place.
This does not modify the original set A.
Correct answers: (a), (c).

4. Explanation: (a, c, d)

(a) 'z' in d
This checks if 'z' is a key in the dictionary.
It returns False safely, without error.
This is safe.

(b) d['z']
This tries to access the value for the key 'z'.
Since 'z' is not in d, this raises a KeyError.
This is not safe.

(c) d.get('z') is None
The get() method safely returns None if the key doesn't exist.
This does not raise an error.
This is safe.

(d) d.setdefault('z', 0) == 0
The setdefault() method checks if 'z' exists. If not, it adds 'z': 0 to the dictionary and returns 0.
This is safe and does not raise an error.
This is safe.
Correct options: (a), (c) and (d)

5. Explanation: (a, b)

(a) x & y== 0
x & y = 0101 & 0010=0000
So, x & y == 0 → True
This returns True

(b) x | y > x
x | y = 0101 | 0010 = 0111 = 7
7 > 5 → True
This returns True

(c) x ^ y < x + y
x ^ y = 0101 ^ 0010 = 0111 = 7
x + y = 5 + 2 = 7
7 < 7 → False
This returns False

(d) not (x and y)
x and y evaluates to y (because both are non-zero, it returns the second operand) → 2 False
This returns False
Correct options: (a), (b)

6. Explanation: (a, b, c)

(a) s1 == s2
Both s1 and s2 contain the same string content "gate"
This compares values.
True

(b) s1 is s2
Python may intern small strings, so both point to the same memory location.
For short strings like "gate", interning happens.
True

(c) s1 == s3
s3 is dynamically created but has the same content "gate".
== True
compares values.
(d) s1 is s3
s3 is created at runtime using join(), so it has the same content but a different memory reference. is checks identity, not value.
False
Correct answers: (a), (b), (c)

7. Explanation: (a, b, d)
Now evaluating each statement:

(a) x is y
True y = x assigns the same memory reference. Both variables point to the same list object.

(b) x == y
True - since both point to the same object, their values are the same.

(c) x = [10, 20]
False — after the append operation, x becomes [10, 20, 30].

(d) len(x) == 3
True - x contains three elements: [10, 20, 30].
Final correct options: (a), (b), (d)

Numerical Answer Type Questions (NAT)

1. Explanation: (243)

Step-by-step Execution:

Call: power (3, 5)

- Since 5 is odd:

→ = 3 power (3* 3, (5 - 1) // 2)

→ = 3 * power (9, 2)

Now evaluate power (9, 2):

- 2 is even:

→ = power (9* 9, 2 // 2)

→ = power (81, 1)

Now evaluate power (81, 1):

1 is odd:

→ = 81 * power (81 * 81, (1 - 1) // 2)

→ = 81 * power (6561, 0)

Now evaluate power (6561, 0):

- 0 is base case:

→ = 1

Now go back:

- power (81, 1) = 81 * 1 = 81
- power (9, 2) = 81
- power (3, 5) = 3 * 81 = 243

Final Answer: 243.

2. Explanation: (65)

Step-by-step Evaluation:

Call: compute(13, 5)

13 is odd →

= 5 + compute(12, 5)

12 is even →

=5+ compute(6, 10)

6 is even →

= 5 + compute (3, 20)

3 is odd →

= 5 + 20+ compute (2, 20)

2 is even →

= 5 + 20+ compute(1, 40)

1 is odd →

= 5 + 20+ 40+ compute(0, 40)

0 → base case → returns 0

Now add them all:

5 +20 + 40 = 65

Final Answer: 65

3. Explanation: (7)

- The count variable increments only when n > 1.
- That means it counts how many non-base recursive calls are made.

Let's trace the call tree for recursive(5):

```
recursive(5)
├── count += 1 → count = 1
├── recursive(4)
│   ├── count += 1 → count = 2
│   ├── recursive(3)
│   │   ├── count += 1 → count = 3
│   │   ├── recursive(2)
│   │   │   ├── count += 1 → count = 4
│   │   │   ├── recursive(1) → base → no increment
│   │   │   └── recursive(0) → base → no increment
│   │   └── recursive(1) → base
│   └── recursive(2)
│       ├── count += 1 → count = 5
│       ├── recursive(1) → base
│       └── recursive(0) → base
└── recursive(3)
    ├── count += 1 → count = 6
    ├── recursive(2)
    │   ├── count += 1 → count = 7
    │   ├── recursive(1) → base
    │   └── recursive(0) → base
    └── recursive(1) → base
```

Summary:

- recursive (5) causes these non-base recursive calls:
- recursive(5) → increments
- recursive(4) → increments
- recursive(3) → increments
- recursive(2)→ increments
- recursive (2) again → increments
- recursive(3) again → increments
- recursive(2) again → increments

↓

- Total increments of count: 7

Final Answer: 7

4. Explanation: (3)

Step-by-step Execution:

1. a.union(b)

Combines all unique elements from both sets a and b:

{10, 20, 30, 40} U {30, 50, 60} = {10, 20, 30, 40, 50, 60}

2. Now intersect with {20, 30, 50}

{10, 20, 30, 40, 50, 60} n (20, 30, 50} = {20, 30, 50}

Final Result:

C = {20, 30, 50}

So, number of elements in c = 3

Final Answer: 3.

5. Explanation: (69)

Step-by-step Execution:

s[::-2] means:

- Start from the end of the string (-1 direction).
- Take every second character in reverse order.

Original string:

S="GATEExam2025"

Index positions:

0 → G

1 → A

2 → T

3 → E
4 → E
5 → x
6 → a
7 → m
8 → 2
0 → 6
10 → 2
11 → 5
Now s[::-2] gives:
Start at index -1 → '5'
Then -3 → '2'
- Then -5 → '0'
Then -7 → 'm'
Then -9 → 'x'
- Then -11 → 'E'

So output = '520mxE'
Last character in the output:
'E'
Now get its ASCII value:
python
ord('E') = 69
Final Answer: 69

6. Explanation: (3)

1. Initial list a:
[1, 2, 2, 3, 4, 4, 5]
2. Convert to a set b = set(a)
Removes duplicates:
b = {1, 2, 3, 4, 5}
3. Convert back to list c = list(b)
c will have the same elements as set b but the order is not guaranteed at this step.
4. Sort the list:
c.sort() → c [1, 2, 3, 4, 5]
5. Access c[2] → third element (0-based indexing):
C[2] = 3

Final Answer: 3

7. Explanation: (3)

Step-by-step Execution:
Step 1: Construct Set A
A = {i for i in range(1, 30) if i % 2==0}
→ Even numbers from 1 to 29
A = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28}
Step 2: Construct Set B
B = {i for i in range(10, 50) if i % 3 == 0}
→ Multiples of 3 from 10 to 49
B = {12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48}
Step 3: Intersection C = A & B
Common elements in both A and B:
C = {12, 18, 24}
Final Step: Count elements in C
len (C) = 3.
Final Answer: 3

8. Explanation: (150)

Step 1: Create Set A (Even numbers from 1 to 50)
A = {2, 4, 6, 8, 10, 12, ..., 50}
These are all numbers divisible by 2 from 1 to 50
So, A contains:
{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50}
Step 2: Create Set B (Multiples of 5 from 1 to 50)
B = {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}
Step 3: Intersection C = A ∩ B
Common elements in A and B:
C = {10, 20, 30, 40, 50}
Step 4: Sum of Elements in C
10 + 20 + 30 + 40 + 50 = 150
Final Answer: 150.

9. Explanation: (7)

Let's trace the loop:

Iteration	n	n // 2	count
1	128	64	1
2	64	32	2
3	32	16	3
4	16	8	4
5	8	4	5
6	4	2	6
7	2	1	7

Now n = 1, and the condition n > 1 becomes false, so the loop stops.
Final Answer: 7

10. Explanation: (36)

We are generating numbers from 1 to 19 that are divisible by both 2 and 3, i.e., divisible by 6.
Let's list those numbers:
- 6
- 12
- 18

These are the numbers between 1 and 19 that satisfy x % 2 ==0 and x % 3==0.
Sum:
6+12+18= 36
Final Answer: 36

Unit 5.2: Stack, Queue and Linked List

[PREVIOUS YEAR EXAM QUESTIONS]

Multiple Choice Questions (MCQ)

[2024: 2 Marks]

1. Match the items in Column 1 with the items in Column 2 in the following table:

Column 1	Column 2
(p) First In First Out	(i) Stacks
(q) Lookup Operation	(ii) Queues
(r) Last In First Out	(iii) Hash Tables

(a) (p) – (ii), (q) – (iii), (r) – (i)
(b) (p) – (ii), (q) – (i), (r) – (iii)
(c) (p) – (i), (q) – (ii), (r) – (iii)
(d) (p) – (i), (q) – (iii), (r) – (ii)

Numerical Answer Type (NAT)

[2024: 1 Mark]

1. The fundamental operations in a double-ended queue D are:

insertFirst(e) – Insert a new element e at the beginning of D.
insertLast(e) – Insert a new element e at the end of D.
removeFirst() – Remove and return the first element of D.
removeLast() – Remove and return the last element of D.
In an empty double-ended queue, the following operations are performed:

```
insertFirst(10)
    insertLast(32)
    a ← removeFirst()
    insertLast(28)
    insertLast(17)
    a ← removeFirst()
    a ← removeLast()
```

The value of a is _____.

[2025: 2 Marks]

2. Consider the following pseudocode:

```
Create empty stack S
    Set x = 0, flag = 0, sum = 0
    Push x onto S
    while (S is not empty){
      if (flag equals 0){
        Set x = x + 1
        Push x onto S
      }
      if (x equals 8){
        Set flag = 1
      }
      if (flag equals 1){
        x = Pop(S)
        if (x is odd):
          Pop(S)
        Set sum = sum + x
      }
    }
    Output sum
```

The value of sum output by a program executing the above pseudocode is _____ (Answer in integer)

ANSWER KEY	
Multiple Choice Questions (MCQ)	
1	
(a)	
Numerical Answer Type (NAT)	
1	**2**
(17 to 17)	(24 to 24)

[PRACTICE QUESTIONS]

Multiple Choice Questions (MCQ)

1. Consider the following statements:

(i) First-in-first out types of computations are efficiently supported by STACKS.
(ii) Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.]
(iii) Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.
(iv) Last-in-first-out type of computations are efficiently supported by QUEUES.

Which of the following is correct?

(a) (ii) and (iii) are true (b) (i) and (ii) are true
(c) (iii) and (iv) are true (d) (ii) and (iv) are true

2. A priority queue Q is used to implement a stack that stores characters. PUSH (C) is implemented as INSERT (Q, C, K) where K is an appropriate integer key chosen by the implementation. POP is implemented as DELETEMIN (Q). For a sequence of operations, the keys chosen are in

(a) Non-increasing order
(b) Non-decreasing order
(c) strictly increasing order
(d) strictly decreasing order

3. Which of the following is essential for converting an infix expression to the postfix form efficiently?

(a) An operator stack
(b) An operand stack
(c) An operand stack and an operator stack
(d) A parse tree