GATE
Since 2004

GATE ACADEMY

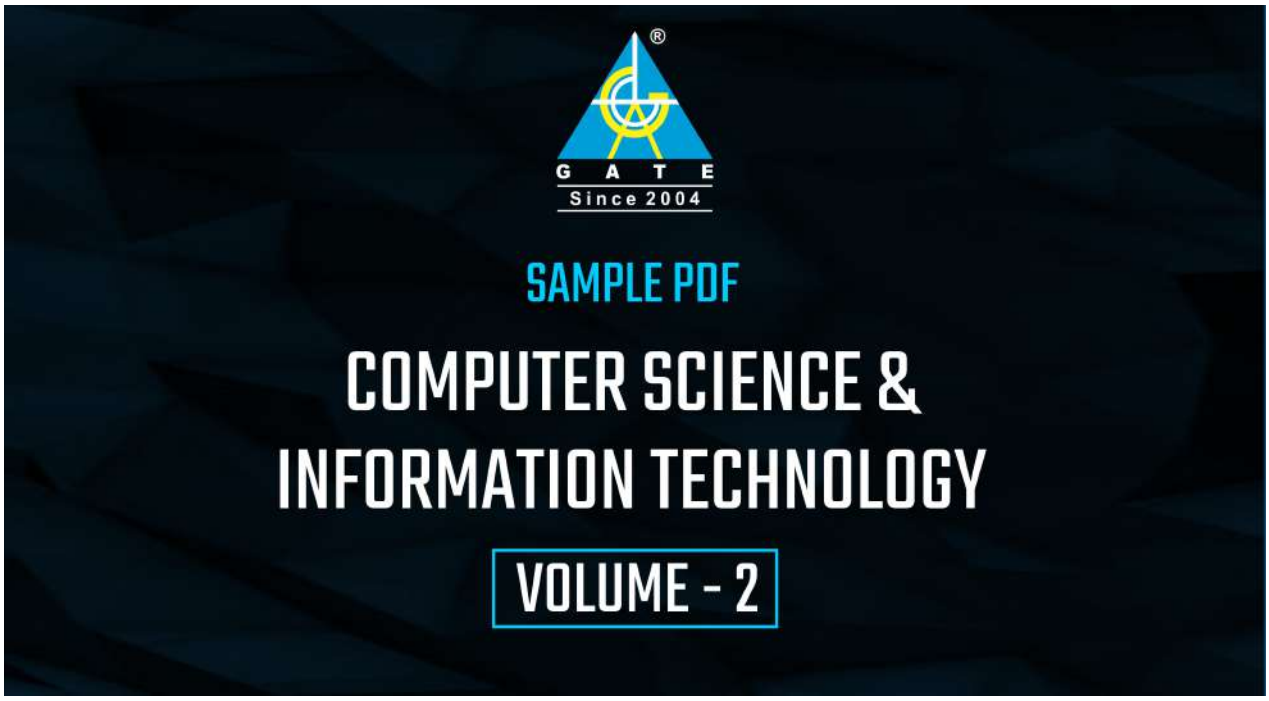
PRESENTS

MOST AWAITED BOOK FOR

GATE – 2022

COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Visit – www.gateacademy.shop



This sample PDF of **GATE Previous Years Solution Book** contains randomly selected questions with solutions from some of the chapters of every subject to let the aspirants have an idea about the content, style and appearance of the book.

Previous Years marks distribution analysis is also given in tabular form with index page of every subject, which contains analysis of GATE papers from 2003 onwards as GATE pattern has turned objective since 2003.

GATE ACADEMY PUBLICATIONS®

35 YEARS GATE SOLVED PAPERS

Best Book for GATE Exam

GATE 2022

COMPUTER SCIENCE
INFORMATION TECHNOLOGY

Topic - Wise Previous 35 Years GATE Questions
With Detailed Solutions

Volume - II

- ☑ Diligent solution of GATE previous year questions (1987-2021)
- ☑ Multi method approach for a single problem to develop crystal clear concepts
- ☑ This book is also a value addition for PSUs/ISRO/DRDO/BARC/DMRC/UGC-NET/State-AE Exams

GATE Since 2004
GATE ACADEMY®
steps to success...

GATE 2022

Computer Science Information Technology

Topic - Wise Previous 35 Years Gate Questions With Detailed Solutions

Live Classroom Program

- Online Live Classroom program by top notch faculties of team GATE ACADEMY, the exact feel of offline Classroom being in your comfort zone.
- Available only on Android app (Gate Academy Live)
- You can also watch it after live class as video on demand (Recorded Video) within your given time.
- You can ask any doubt by using chat box during the live class.
- All Study material will be provided in digital/ printed format.

Video on Demand

- VoD is the best online tool for self preparing students.
- Available for Android and Windows users.
- All you need is decent internet connection and our GATE ACADEMY Learning App.
- All Study material will be provided in digital/ printed format.
- Dedicated and detailed discussion with doubt clearing portal managed by our top notch faculties and experts.

Pen Drive & G-Drive Course

- It is best offline mode of video course. (No Internet Required for PD Course, for GD Course Internet is required for downloading the lecture)
- Dedicated and detailed discussion with doubt clearing portal managed by our top notch faculties and experts.
- Complete packaged course with all Study materials.
- User Friendly technology and Ready to ship course.

Tablet Course

- Tablet course will be delivered through Samsung Galaxy Tab A (WiFi/Cellular).
- All GATE ACADEMY apps will be pre loaded.
- Dedicated discussion and doubt portal managed by our faculties and experts.
- The course content will be exactly same as VOD course but you do not need Internet connection to play the video (as it is an offline course)

GATE ACADEMY Learning Pvt. Ltd.

STREET - 04, NARSINGH VIHAR, KATULBOD, BHILAI, CHHATTISGARH (490022)

CALL : 97131 13156, 0788-4034176 Email : pdcourse@gateacademy.co.in

9789384000691

₹ 1000/-

GATE-CS/IT-691-2

Video Tutorial (Youtube)
Channel Name
GATE ACADEMY - CS IT
Channel Link
https://www.youtube.com/channel/UCNGPqc8IajHf8yUnFtFJ4Vg

Join
GATE ACADEMY - CS IT
Channel Link
https://t.me/gateacademyCS

Video Tutorial (Youtube)
Channel Name
GATE ACADEMY CS IT: UGC NET/JRF/NIC/Gov Exam
Channel Link
https://www.youtube.com/channel/UCNburME-Ec74bow9hhvASRg

GATE ACADEMY®
steps to success...

GATE 2022

Computer Science & Information Technology

(Volume-II)

TOPIC WISE GATE SOLUTIONS

1987-2021

Team GATE ACADEMY

Topic wise GATE Solutions

Computer Science & Information Technology

Volume - II

Cover Design : Gajendra Banjare

Type & Setting : Rajendra Banjare

Book Code : GATE-CS/IT-691-2

First Edition : August 2021

ISBN : 978-93-84000-69-1

GATE ACADEMY Learning Pvt. Ltd

Ground Floor, A-115, Shopping Complex, Smriti Nagar,

Bhilai, Chhattisgarh 490020

Phone : 0788 - 4034176

Help Desk No. - +91-97131-13156

For Feedback & Suggestions...

galpl.bhilai@gmail.com

Price ₹ : 1000/-

IMPORTANCE of GATE

GATE examination has been emerging as one of the most prestigious competitive exam for engineers. Earlier it was considered to be an exam just for eligibility for pursuing PG courses, but now GATE exam has gained a lot of attention of students as this exam open an ocean of possibilities like :

1. **Admission into IISc, IITs, IIITs, NITs**
   A good GATE score is helpful for getting admission into IISc, IITs, IIITs, NITs and many other renowned institutions for M.Tech./M.E./M.S. An M.Tech graduate has a number of career opportunities in research fields and education industries. Students get ₹ 12,400 per month as stipend during their course.
2. **Selection in various Public Sector Undertakings** (**PSUs**)
   A good GATE score is helpful for getting job in government-owned corporations termed as **Public Sector Undertakings** (**PSUs**) in India like IOCL, BHEL, NTPC, BARC, ONGC, PGCIL, DVC, HPCL, GAIL, SAIL & many more.
3. 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.
4. **Foreign universities through GATE**
   GATE has crossed the boundaries to become an international level test for entry into postgraduate engineering programmes in abroad. Some institutes in two countries **Singapore** and **Germany** are known to accept GATE score for admission to their PG engineering programmes.
5. **National Institute of Industrial Engg. (NITIE)**
   - NITIE offers **PGDIE / PGDMM / PGDPM** on the basis of GATE scores. The shortlisted candidates are then called for group Discussion and Personal Interview rounds.
   - NITIE offers a Doctoral Level Fellowship Programme recognized by Ministry of HRD (MHRD) as equivalent to PhD of any Indian University.
   - Regular full time candidates those who will qualify for the financial assistance will receive ₹ 25,000 during 1st and 2nd year of the Fellowship programme and ₹ 28,000 during 3rd, 4th and 5th year of the Fellowship programme as per MHRD guidelines.
6. **Ph.D. in IISc/ IITs**
   - IISc and IITs take admissions for Ph.D. on the basis of GATE score.
   - Earn a Ph.D. degree directly after Bachelor's degree through integrated programme.
   - A fulltime residential researcher (RR) programme.
7. **Fellowship Program in management (FPM)**
   - Enrolment through GATE score card
   - Stipend of ₹ 22,000 – 30,000 per month + HRA
   - It is a fellowship program
   - Application form is generally available in month of sept. and oct.

Note : In near future, hopefully GATE exam will become a mandatory exit test for all engineering students, so take this exam seriously. Best of LUCK !

GATE Exam Pattern

Section	Number of Questions	Marks Per Question	Total Marks
General Aptitude	5	1	5
	5	2	10
Technical + Engineering Mathematics	25	1	25
	30	2	60
Total Duration : 3 hours	**Total Questions : 65**		**Total Marks : 100**

Note :

(i) 40 to 45 marks will be allotted to Numerical Answer Type Questions.

(ii) MSQ also added from GATE 2021 for which **no negative** marking.

Pattern of Questions :

GATE 2021 would 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 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 mark 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.

Example : If the wire diameter of a compressive helical spring is increased by 2%, the change in spring stiffness (in %) is _ (correct to two decimal places).

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

GATE SYLLABUS

Section 1 : Engineering Mathematics

Discrete Mathematics: Propositional and first order logic. Sets, relations, functions, partial orders and lattices. Monoids, Groups. Graphs: connectivity, matching, coloring. Combinatorics: counting, recurrence relations, generating functions.

Linear Algebra: Matrices, determinants, system of linear equations, eigenvalues and eigenvectors, LU decomposition.

Calculus: Limits, continuity and differentiability. Maxima and minima. Mean value theorem. Integration.

Probability and Statistics: Random variables. Uniform, normal, exponential, poisson and binomial distributions. Mean, median, mode and standard deviation. Conditional probability and Bayes theorem.

Section 2 : Digital Logic

Boolean algebra. Combinational and sequential circuits. Minimization. Number representations and computer arithmetic (fixed and floating point).

Section 3 : Computer Organization and Architecture

Machine instructions and addressing modes. ALU, data-path and control unit. Instruction pipelining, pipeline hazards. Memory hierarchy: cache, main memory and secondary storage; I/O interface (interrupt and DMA mode).

Section 4 : Programming and Data Structures

Programming in C. Recursion. Arrays, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs.

Section 5 : Algorithms

Searching, sorting, hashing. Asymptotic worst case time and space complexity. Algorithm design techniques: greedy, dynamic programming and divide-and-conquer. Graph traversals, minimum spanning trees, shortest paths.

Section 6 : Theory of Computation

Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and contex-free languages, pumping lemma. Turing machines and undecidability.

Section 7 : Compiler Design

Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimisation, Data flow analysis: constant propagation, liveness analysis, common subexpression elimination.

Section 8 : Operating System

System calls, processes, threads, inter-process communication, concurrency and synchronization. Deadlock. CPU and I/O scheduling. Memory management and virtual memory. File systems.

Section 9 : Databases

ER-model. Relational model: relational algebra, tuple calculus, SQL. Integrity constraints, normal forms. File organization, indexing (e.g., B and B+ trees). Transactions and concurrency control.

Section 10 : Computer Networks

Concept of layering: OSI and TCP/IP Protocol Stacks; Basics of packet, circuit and virtual circuit-switching; Data link layer: framing, error detection, Medium Access Control, Ethernet bridging; Routing protocols: shortest path, flooding, distance vector and link state routing; Fragmentation and IP addressing, IPv4, CIDR notation, Basics of IP support protocols (ARP, DHCP, ICMP), Network Address Translation (NAT); Transport layer: flow control and congestion control, UDP, TCP, sockets; Application layer protocols: DNS, SMTP, HTTP, FTP, Email.

General Aptitude (GA) :

Verbal Aptitude :

Basic English grammar: tenses, articles, adjectives, prepositions, conjunctions, verb-noun agreement, and other parts of speech Basic vocabulary: words, idioms, and phrases in context Reading and comprehension Narrative sequencing.

Quantitative Aptitude :

Data interpretation: data graphs (bar graphs, pie charts, and other graphs representing data), 2- and 3-dimensional plots, maps, and tables Numerical computation and estimation: ratios, percentages, powers, exponents and logarithms, permutations and combinations, and series Mensuration and geometry Elementary statistics and probability.

Analytical Aptitude :

Logic: deduction and induction Analogy, Numerical relations and reasoning.

Spatial Aptitude :

Transformation of shapes: translation, rotation, scaling, mirroring, assembling, and grouping Paper folding, cutting, and patterns in 2 and 3 dimensions.

PREFACE

It is our pleasure, that we insist on presenting **"GATE 2022 Computer Science & Information Technology (Volume - II)"** authored for Computer Science and Engineering to all of the aspirants and career seekers. The prime objective of this book is to respond to tremendous amount of ever growing demand for error free, flawless and succinct but conceptually empowered solutions to all the question over the period 1987 - 2021.

This book serves to the best supplement the texts for Computer Science Engineering but shall be useful to a larger extent for Information Technology & other National Level CS related Exams as well. Simultaneously having its salient feature the book comprises :

- Step by step solution to all questions.
- Complete analysis of questions, i.e. chapter wise as well as year wise.
- Detailed explanation of all the questions.
- Solutions are presented in simple and easily understandable language.
- It covers all GATE questions from 1987 to 2021 (35 years).

The authors do not sense any deficit in believing that this title will in many aspects, be different from the similar titles within the search of student.

We would like to express our sincere appreciation to **Mrs. Sakshi Dhande Ma'am** (Co-founder, GATE ACADEMY Group) for her constant support and constructive suggestions and comments in reviewing the script.

In particular, we wish to thank GATE ACADEMY expert team members for their hard work and consistency while designing the script.

The final manuscript has been prepared with utmost care. However, going a line that, there is always room for improvement in anything done, we would welcome and greatly appreciate the suggestions and corrections for further improvement.

Umesh Dhande

(Director, GATE ACADEMY Learning)

ACKNOWLEDGEMENT

Years of recurring effort went into the volume which is now ready to cover the aptitude of GATE aspirants.

We are glad of this opportunity to acknowledge the views and to express with all the weaknesses of mere words the gratitude that we must always feel for the generosity of them.

We now express our gracious gratitude to the persons who have contributed a lot in order to put forth this into device. They are to be mentioned here and they are Nutesh, Yogesh, Shama, Ankit, Faizan and Firdous.

We would also like to express our gracious gratitude to the faculty members of GATE ACADEMY who has contributed a lot in order to put forth this into device. They are to be mentioned here and they are **Gurupal S. Chawla, Ajay Das, Sateesh Kesharwani, Kavindra Krishna, Sachin Tanwar** and **Aishwarya Vijay.**

Lastly, we take this opportunity to acknowledge the service of the total team of publication and everyone who collaborated in producing this work.

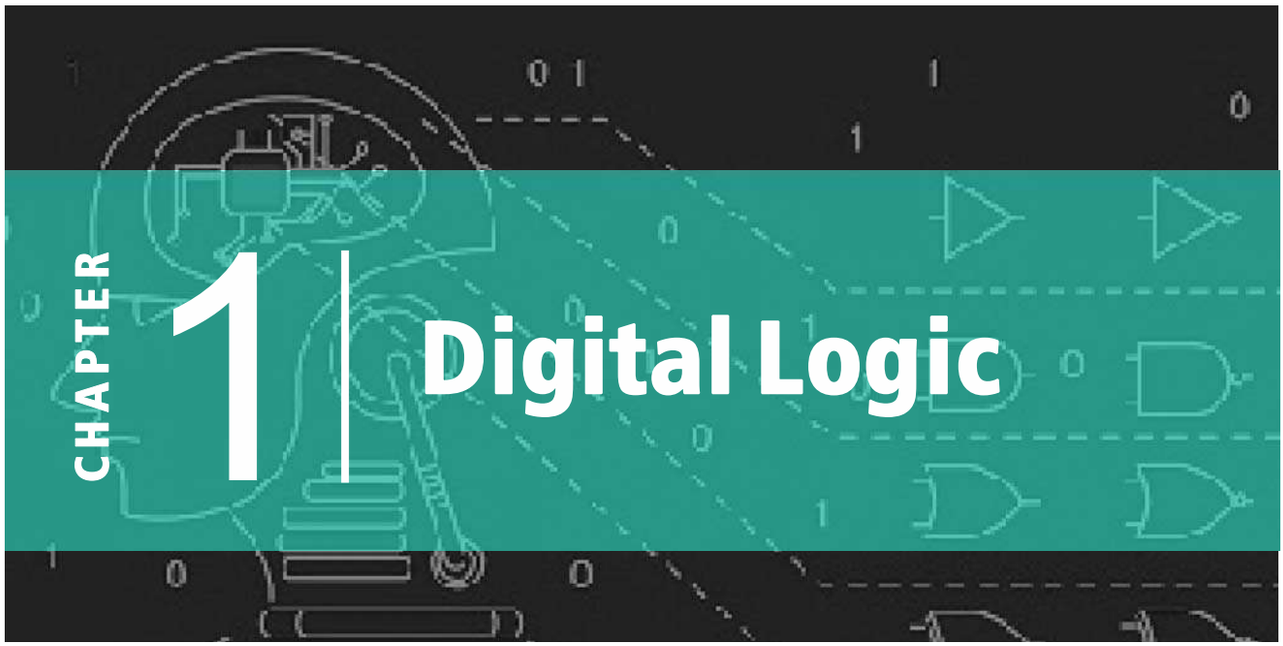
GATE ACADEMY

COMPUTER SCIENCE & INFORMATION TECHNOLOGY

VOLUME – 2

1. DIGITAL LOGIC
2. COMPUTER ORGANIZATION & ARCHITECTURE
3. OPERATING SYSTEM
4. THEORY OF COMPUTATION
5. COMPILER DESIGN
6. COMPUTER NETWORKS

Visit – www.gateacademy.shop



CHAPTER 1 Digital Logic

Marks Distribution of Digital Logic in Previous Year GATE Papers.

Exam Year	1 Mark Ques.	2 Marks Ques.	Total Marks
2003	1	4	9
2004*	4	5	14
2005*	4	4	12
2006*	1	5	11
2007*	3	5	13
2008*	4	1	6
2009	2	-	2
2010	3	2	7
2011	2	3	8
2012	2	-	2
2013	3	-	3
2014 Set-1	2	1	4
2014 Set-2	3	1	5
2014 Set-3	2	2	6

Exam Year	1 Mark Ques.	2 Mark Ques.	Total Marks
2015 Set-1	1	2	5
2015 Set-2	1	2	5
2015 Set-3	-	3	6
2016 Set-1	3	2	7
2016 Set-2	3	-	3
2017 Set-1	2	1	4
2017 Set-2	1	3	7
2018	2	2	6
2019	3	2	7
2020	-	2	4
2021 Set-1	1	2	5
2021 Set-2	2	2	6

*** CS and IT combined**

Syllabus : Digital Logic

Boolean algebra. Combinational and sequential circuits. Minimization. Number representations and computer arithmetic (fixed and floating point).

Contents : Digital Logic

1990 IISc Bangalore

1 Show with the help of a block diagram how the Boolean function : $f = AB + BC + CA$ can be realized using only a 4:1 multiplexer.

1991 IIT Madras

2 Consider the number given by the decimal expression.

$$16^3 \times 9 + 16^2 \times 7 + 16 \times 5 + 3$$

The number of 1's in the unsigned binary representation of the number is _____.

1999 IIT Bombay

3 Booth's coding in 8 bits for the decimal number –57 is

(A) $0-100+1000$

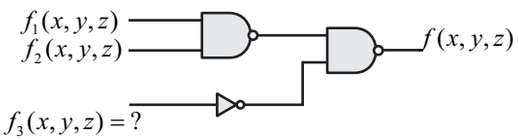
(B) $0-100+100-1$

(C) $0-1+100-10+1$

(D) $00-10+100-1$

2002 IISc Bangalore

4 Consider the following logic circuit whose inputs are functions f_1, f_2, f_3 and output is f.



Given that

$f_1(x,y,z) = \Sigma(0,1,3,5)$

$f_2(x,y,z) = \Sigma(6,7)$ and

$f(x,y,z) = \Sigma(1,4,5)$, f_3 is

(A) $\Sigma(1, 4, 5)$

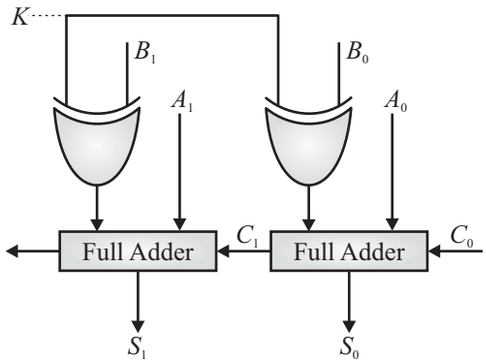
(B) $\Sigma(6, 7)$

(C) $\Sigma(0, 1, 3, 5)$

(D) None of the above

2003 IIT Madras

5 Consider the ALU shown below



If the operands are in 2's complement representation, which of the following operations can be performed by suitably setting the control lines K and C_0 only (+ and – denote addition and subtraction respectively)?

(A) A + B, and A – B, but not A + 1

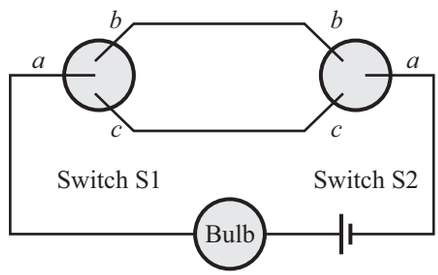
(B) A + B, and A + 1, but not A – B

(C) A + B, but not A – B, or A + 1

(D) A + B, A – B and A + 1

2004 IIT Delhi

6 A two-way switch has three terminals *a*, *b* and *c*. In ON position (logic value 1), *a* is connected to *b*, and in OFF position, *a* is connected to *c*. Two of these two-way switches *S*1 and *S*2 are connected to a bulb as shown below.



Which of the following expression, if true, will always result in the lighting of the bulb?

(A) $S1 \cdot \overline{S2}$ (B) $S1 + S2$

(C) $\overline{S1 \oplus S2}$ (D) $S1 \oplus S2$

7 A circuit outputs a digit in the form of 4 bits. 0 is represented by 0000, 1 by 0001,……, 9 by 1001. A combinational circuit is to be designed which takes these 4 bits as input and output 1 if the digit ≥ 5, and 0 otherwise. If only AND, OR and NOT gates may be used what is the minimum number of gates required?

(A) 2 (B) 3

(C) 4 (D) 5

2007 IIT Kanpur

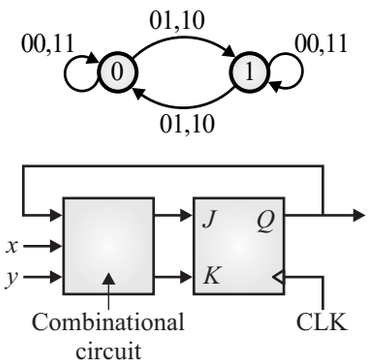
8 Which of the following input sequences for a cross-coupled R-S flip-flop realized with two NAND gates may lead to an oscillation?

(A) 11, 00 (B) 01, 10

(C) 10, 01 (D) 00, 11

2008 IISc Bangalore

9 Consider the following state diagram and its realization by a JK flip flop.



The combinational circuit generates J and K in terms of x, y and Q. The Boolean expressions for J and K are :

(A) $\overline{x \oplus y}$ and $\overline{x \oplus y}$

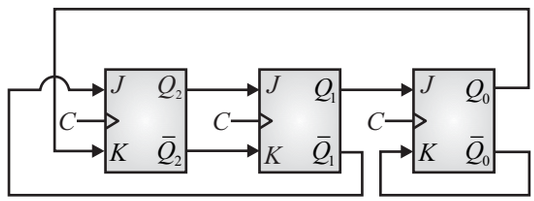
(B) $\overline{x \oplus y}$ and $x \oplus y$

(C) $x \oplus y$ and $\overline{x \oplus y}$

(D) $x \oplus y$ and $x \oplus y$

2014 IIT Kharagpur

10



The above synchronous sequential circuit built using JK flip-flops is initialized with $Q_2Q_1Q_0 = 000$. The state sequence for this circuit for the next 3 clock cycles (C) is _____.

(A) 001, 010, 011 (B) 111, 110, 10

(C) 100, 110, 111 (D) 100, 011, 001

Answers Digital Logic

1.	*	2.	9	3.	B	4.	A	5.	A
6.	C	7.	B	8.	D	9.	D	10.	C

Explanations Digital Logic

1 *

Given : 3-variable Boolean function as,

$$f = AB + BC + CA \qquad \ldots(i)$$

Method 1

Converting equation (i) into standard canonical SOP form as,

$$f = AB(C + \bar{C}) + BC(A + \bar{A}) + CA(B + \bar{B})$$

$$f = \underbrace{\bar{A}BC}_{3} + \underbrace{A\bar{B}C}_{5} + \underbrace{AB\bar{C}}_{6} + \underbrace{ABC}_{7}$$

$$f = \Sigma m(3, 5, 6, 7) \qquad \ldots(ii)$$

According to question, we have 4×1 MUX, it requires only 2-selection lines, but we have 3-variable A, B, C in which A and B can used as selection line of 4×1 MUX and remaining C can be used as input for 4×1 MUX and selection of that input (C) as follows,

	I_0	I_1	I_2	I_3
$\bar{C}$	0	2	4	[6]
C	1	[3]	[5]	[7]
	0	C	C	1

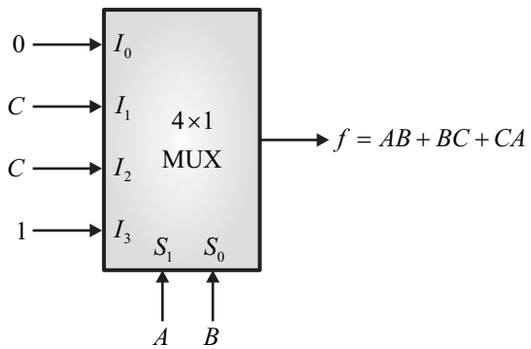
$\rightarrow$ Input of 4×1 MUX

Here, A and $B \rightarrow$ Selection line S_1 and S_2 respectively.

From above table, input of 4×1 MUX,

$$I_0 = 0,\ I_1 = C,\ I_2 = C,\ I_3 = 1$$

Finally, 4×1 MUX circuit with inputs and selection lines as,



Hence, the above circuit shows the realization of $f = AB + BC + CA$ using 4×1 MUX.

Method 2

Converting equation (i), into standard canonical SOP form as,

$$f = AB(C + \bar{C}) + BC(A + \bar{A}) + CA(B + \bar{B})$$

$$f = \underbrace{\bar{A}BC}_{3} + \underbrace{A\bar{B}C}_{5} + \underbrace{AB\bar{C}}_{6} + \underbrace{ABC}_{7}$$

$$f = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$f = \bar{A}BC + A\bar{B}C + AB(\bar{C} + C) \quad (\because\ \bar{C} + C = 1)$$

$$f = \bar{A}BC + A\bar{B}C + AB$$

Adding $\bar{A}\bar{B}\cdot 0$ in above equation,

$$f = \bar{A}\,\bar{B}\cdot 0 + \bar{A}B\cdot C + A\bar{B}\cdot C + AB\cdot 1 \quad \ldots(ii)$$

If A and B work as selection lines S_1, S_0 respectively and I_0, I_1, I_2, I_3 are input of 4×1 MUX , then

$S_1S_0 = \bar{A}\,\bar{B} \rightarrow$ Select input I_0

$S_1S_2 = \bar{A}B \rightarrow$ Select input I_1

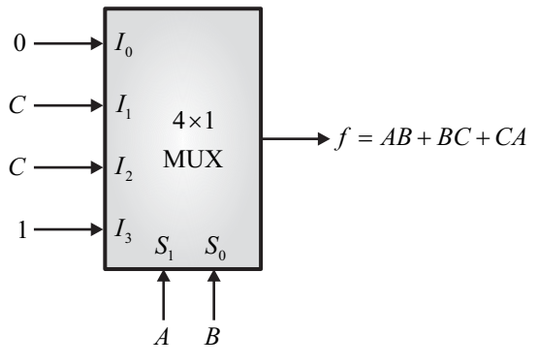
$S_1S_2 = A\bar{B} \rightarrow$ Select input I_2

$S_1S_2 = AB \rightarrow$ Select input I_3

So, equation (ii) becomes as,

$$f = \bar{A}\,\bar{B}\cdot \underset{\downarrow}{0} + \bar{A}B\cdot \underset{\downarrow}{C} + A\bar{B}\cdot \underset{\downarrow}{C} + AB\cdot \underset{\downarrow}{1}$$
$$I_0 \qquad I_1 \qquad I_2 \qquad I_3$$

So, 4×1 MUX can be designed for function (f) is,



Hence, the above circuit shows the realization of $f = AB + BC + CA$ using 4×1 MUX.

Key Point

(i) MUX is a universal combinational logic circuit, which can implement any logic gate.

(ii) MUX is also called as parallel to serial converter device.

(iii) The output of MUX generally written in SOP form.

2 9

Given : Number in decimal expression is,

$$16^3 \times 9 + 16^2 \times 7 + 16 \times 5 + 3 \quad \ldots(i)$$

Method 1

We can re-write equation (i) as,

$$9 \times 16^3 + 7 \times 16^2 + 5 \times 16^1 + 3 \times 16^0 \quad \ldots(ii)$$

Equation (ii) shows a decimal conversion of Hexadecimal number $(9753)_{16}$.

For conversion of Hexadecimal number into un-signed binary number, each digit of Hexadecimal number converted into 4-bit binary number as,

$$(9)_{16} \rightarrow (1001)_2$$
$$(7)_{16} \rightarrow (0111)_2$$
$$(5)_{16} \rightarrow (0101)_2$$
$$(3)_{16} \rightarrow (0011)_2$$

So, finally unsigned binary equivalent of $(9753)_{16}$ is,

$$(9753)_{16} \xrightarrow[Equivalent]{Binary} \underbrace{(1001\ 0111\ 0101\ 0011)_2}_{\text{Number of 1's = 9}}$$

Hence, the number of 1's in unsigned binary equivalent is **9**.

Method 2

Given number in decimal expression is,

$$\Rightarrow \quad 16^3 \times 9 + 16^2 \times 7 + 16 \times 5 + 3$$
$$\Rightarrow \quad 4096 \times 9 + 256 \times 7 + 16 \times 5 + 3$$
$$\Rightarrow \quad 36,864 + 1792 + 80 + 3$$
$$\Rightarrow \quad (38739)_{10}$$

Now, convert decimal number $(38739)_{10}$ into un-signed binary number as,

			Remainder
2	38739	→	1
2	19369	→	1
2	9684	→	0
2	4842	→	0
2	2421	→	1
2	1210	→	0
2	605	→	1
2	302	→	1
2	151	→	0
2	75	→	1
2	37	→	1
2	18	→	1
2	9	→	0
2	4	→	1
2	2	→	0
	1	→	0

Unsigned binary equivalent is,

$$\underbrace{(1001\ 0111\ 0101\ 0011)_2}_{\text{Number of 1's =9}}$$

Therefore, the number of 1's in unsigned binary equivalent is **9**.

Hence, the correct answer is 9.

Key Point

(i) For conversion of hexadecimal number (i.e. base-16 number) into binary number (i.e. base-2 number), convert each digit of hexadecimal number into 4-bit binary number.

Example :

$(1331)_{16} \rightarrow (0001\ 0011\ 0011\ 0001)_2$

- 4-bit binary of $(1)_{16}$
- 4-bit binary of $(3)_{16}$
- 4-bit binary of $(3)_{16}$
- 4-bit binary of $(1)_{16}$

(ii) For conversion of octal number (i.e. base-8 number) into binary number (i.e. base-2 number), convert each digit of octal number into 3-bit binary number.

Example :

$(1331)_8 \rightarrow (001\ 011\ 011\ 001)_2$

- 3-bit binary of $(1)_8$
- 3-bit binary of $(3)_8$
- 3-bit binary of $(3)_8$
- 3-bit binary of $(1)_8$

3 (B)

Given : Decimal number is, $(-57)_{10}$

(Given number is negative)

Method 1

We take some steps to solve this question as,

Step-1 : First convert $(+57)_{10}$ into 8-bit binary number as,

$$(+57)_{10} \rightarrow (00111001)_2$$

Now, take the 2's complement of (00111001), because given number is negative. So,

00111001

↓2's complement

11000111

So, 2's complement representation of $(-57)_{10}$ is,

$$(-57)_{10} \rightarrow (11000111)_2$$

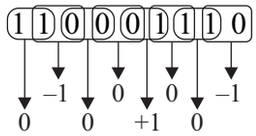
Step-2 : Append zero after LSB in 2's complement representation of $(-57)_{10}$, because of booth algorithm encoding,

$(1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0)_2$

↓

Appended bit

Step-3 : Now, making groups of 2-bit while traversing from LSB to MSB an apply booth encoding for $00 \rightarrow 0$, $01 \rightarrow +1$, $10 \rightarrow -1$ and $11 \rightarrow 0$.



So, $(-57)_{10}$ in 8-bit booth coding is,

$$0-100+100-1$$

Hence, the correct option is (B).

Method 2

Checking from options,

From option (A) : $0-100+1000$

Convert given binary string into decimal number as,

0	– 1	0	0	+ 1	0	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$\Rightarrow \quad 0\times2^7+(-1)\times2^6+0\times2^5+0\times2^4+1\times2^3+0\times2^2+0\times2^1+0\times2^0$$

$$\Rightarrow \quad -64+8$$

$$\Rightarrow \quad (-56)_{10}$$

Thus, it does not satisfy the given decimal number.

Hence, option (A) is incorrect.

From option (B) : $0-100+100-1$

Convert given binary string into decimal number as,

0	– 1	0	0	+ 1	0	0	– 1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$\Rightarrow \quad 0\times2^7+(-1)\times2^6+0\times2^5+0\times2^4+1\times2^3+0\times2^2+0\times2^1+(-1)\times2^0$$

$$\Rightarrow \quad (-64)+8-1$$

$$\Rightarrow \quad (-57)_{10}$$

Thus, it satisfy the given decimal number.

Hence, option (B) is correct.

From option (C) : $0-1+100-10+1$

Convert given binary string into decimal number as,

0	– 1	1	0	0	– 1	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$\Rightarrow \quad 0\times2^7+(-1)\times2^6+1\times2^5+0\times2^4+0\times2^3$$
$$+(-1)\times2^2+0\times2^1+(1)\times2^0$$

$$\Rightarrow \quad (-64)+32-4+1$$

$$\Rightarrow \quad (-35)_{10}$$

Thus, it does not satisfy the given decimal number.

Hence, option (C) is incorrect.

From option (D) : $00-10+100-1$

Convert given binary string into decimal number as,

0	0	– 1	0	+ 1	0	0	– 1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$\Rightarrow \quad 0\times2^7+0\times2^6+(-1)\times2^5+0\times2^4+1\times2^3$$
$$+0\times2^2+0\times2^1+(-1)\times2^0$$

$$\Rightarrow \quad (-32)+8-1$$

$$\Rightarrow \quad (-25)_{10}$$

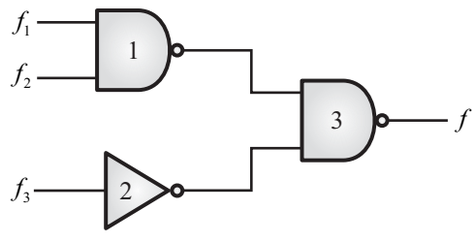
Thus, it does not satisfy the given decimal number.

Hence, option (D) is incorrect.

Hence, the correct option is (B).

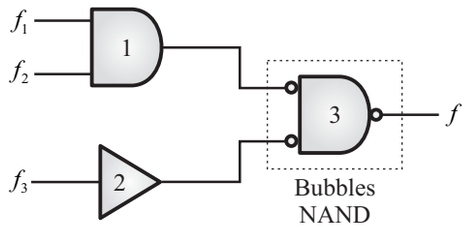
4 (A)

Given : Logic circuit whose input are f_1, f_2 and f_3 are shown below,

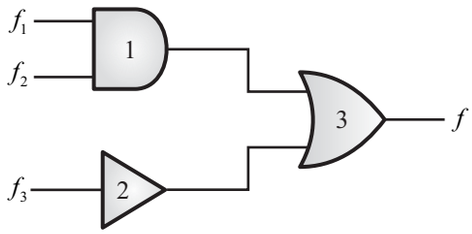


Method 1

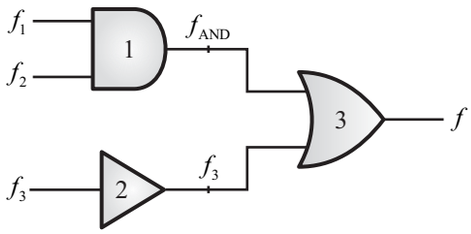
Now, shifting the bubbles of gate-1 and gate-2 towards gate-3 and above circuit becomes,



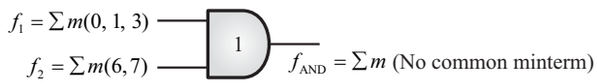
Bubbled NAND gate will work as a OR gate so, replace gate-3 by OR gate, then above circuit becomes,



Gate-2 work as a buffer and buffer is used only for delay and its output is equal to input so,

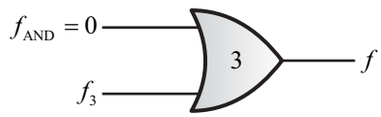


AND gate-1 passes only common minterms of input applied to it so,



Thus, $f_{AND} = \text{Null} = \text{Zero}.$

So, output of OR gate-3 is,



So, $f = f_{AND} + f_3$

$f = 0 + f_3$

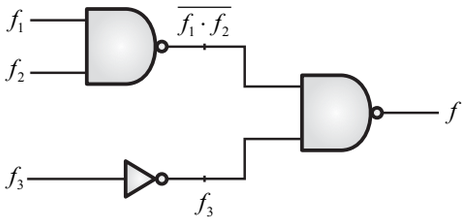
$f = f_3 = \Sigma m(1, 4, 5)$

Thus, $f_3 = \Sigma m(1, 4, 5)$

Hence, the correct option is (A).

Method 2

Given circuit is shown below,



So, $f = \overline{\overline{f_1 \cdot f_2} \cdot \overline{f_3}}$

Apply De-Morgan law,

$$f = \overline{\overline{f_1 \cdot f_2}} + \overline{\overline{f_3}}$$

$$f = f_1 \cdot f_2 + f_3 \quad \ldots(i)$$

Here, $f_1 \cdot f_2$ means common minterms between f_1 and f_2.

i.e. $f_1 \cdot f_2 = \Sigma m(0, 1, 3, 5) \cdot \Sigma m(6, 7)$

$f_1 \cdot f_2 = \text{Null} \rightarrow$ Because no common minterms

So, equation (i) becomes,

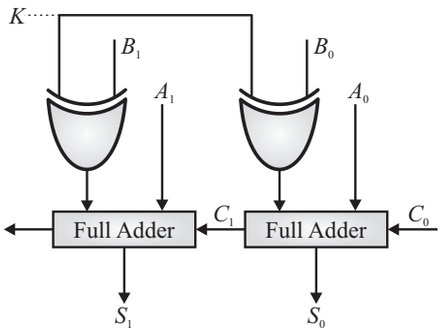
$$f = 0 + f_3$$

So, $f = f_3 = \Sigma m(1, 4, 5)$

Hence, the correct option is (A).

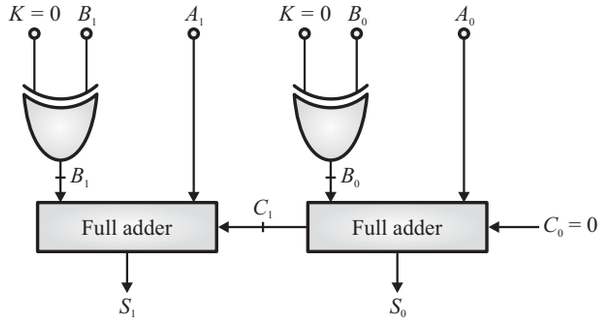
5 (A)

Given : Circuit is shown below,



According to question, $A = A_1A_0$, $B = B_1B_0$ and their sum $S = S_1S_0$ and K, C_0 in our hand for controlling the circuit,

Case 1 : When $K = 0$ and $C_0 = 0$, then circuit becomes,



So, result

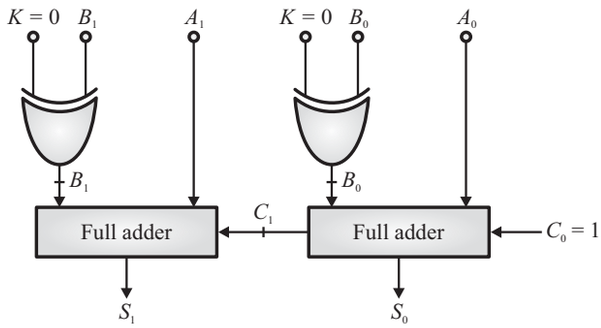
$$\left.\begin{array}{rcc} & C_1 & C_0 = 0 \\ A \Rightarrow & A_1 & A_0 \\ B \Rightarrow & +B_1 & B_0 \\ \hline S \Rightarrow & S_1 & S_0 \end{array}\right\} \text{Addition of } A \text{ and } B$$

So, result $\Rightarrow S = A + B$

Thus, $K = 0 = C_0$

then $S = A + B \Rightarrow$ Addition of A and B.

Case 2 : When $K = 0$ and $C_0 = 1$, then circuit becomes as,



So, result

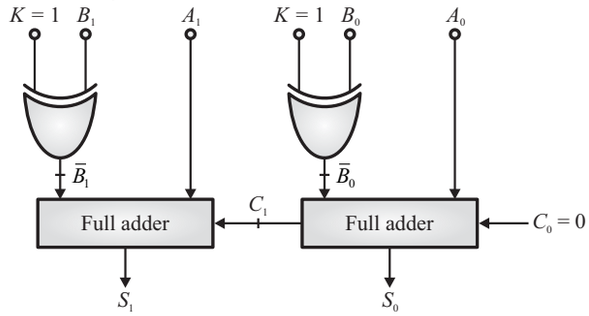
$$\left.\begin{array}{rcc} & C_1 & C_0 = 1 \\ A \Rightarrow & A_1 & A_0 \\ B \Rightarrow & +B_1 & B_0 \\ \hline S \Rightarrow & S_1 & S_0 \end{array}\right\} \begin{array}{l}\text{Addition of } A \text{ and } B \\ \text{with initial carry}\end{array}$$

So, result $\Rightarrow S = (A + B) + C_0 = 1$

Thus, $K = 0$, $C_0 = 1$, then $S = (A + B) + 1$

(Addition with initial carry)

Case 3 : When $K=1$, $C_0=1$, then circuit becomes,



So, result

$$\begin{array}{l} \quad C_1\ C_0=0 \\ A\Rightarrow\ A_1\ A_0 \\ \overline{B}\Rightarrow+\overline{B}_1\ \overline{B}_0 \\ \hline S\Rightarrow\ S_1\ S_0 \end{array} \Big\} 1\text{'s complement of } B$$

So, result $\Rightarrow S=A+\overline{B}$

$S=A+1$'s complement of (B)

So, we can say that,

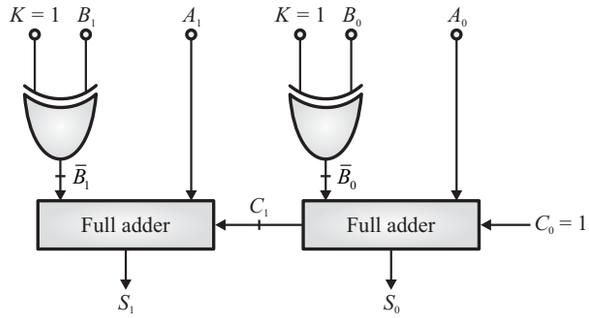
$S=1$'s complement, addition of A and B.

Thus, 1's complement, addition of A and B is nothing but a, subtraction of A and B, using 1's complement technique. So,

$S=1$'s complement addition of A & $B=A-B$

Thus, $K=1$ and $C=0$, then $S=A-B$.

Case 4 : When $K=1, C_0=1$, then circuits becomes as,



So, result,

$$\begin{array}{l} \quad C_1\ C_0=1 \\ A\Rightarrow\ A_1\ A_0 \\ B\Rightarrow+\overline{B}_1\ \overline{B}_0 \\ \hline S\Rightarrow\ S_1\ S_0 \end{array} \Big\} 1\text{'s complement of } B$$

So, result $\Rightarrow S=\underbrace{A+\overline{B}}_{\substack{1\text{'s complement}\\ \text{addition}}}+\underbrace{1}_{\substack{\text{Initial}\\ \text{carry}}}$

$S=(1$'s complement addition of A and $B)+1$

$S=2$'s complement addition of $(A+B)$

Thus, 2's complement addition of A and B is nothing but a subtraction of A and B using 2's complement technique. So we can say that,

$S=2$'s complement addition of A & $B=A-B$

Thus, $K=1$ and $C_0=1$, then $S=A-B$.

Thus, by controlling K and C, given circuit provides three operations,

1. $A+B$
2. $A-B$
3. $(A+B)+1$

Hence, the correct option is (A).

6 (C)

According to question,

$$\text{Switch} \underset{(a\text{ to }b)}{\longrightarrow} \text{ON} \longrightarrow \text{Logic-1}$$

$$\text{Switch} \underset{(a\text{ to }c)}{\longrightarrow} \text{OFF} \longrightarrow \text{Logic-0}$$

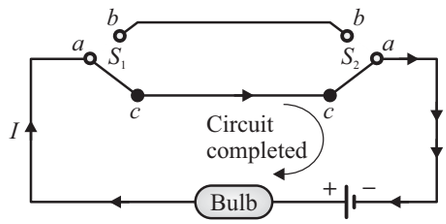
Let us consider 4-cases for analysis of this problem in detailed manner,

Case-1 :

$$\text{Switch}\ (S_1) \underset{(a\text{ to }c)}{\longrightarrow} \text{OFF} \longrightarrow \text{Logic-0}$$

$$\text{Switch}\ (S_2) \underset{(a\text{ to }c)}{\longrightarrow} \text{OFF} \longrightarrow \text{Logic-0}$$

Then circuit becomes as,



Now, it is clear that above circuit have completed a loop and current I flow through the bulb and bulb goes ON/logic-1/glows.

Thus,

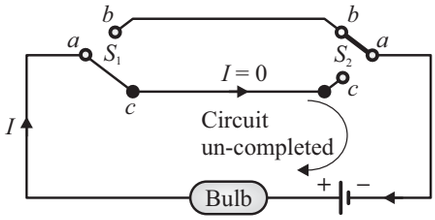
S_1	S_2	Bulb
0	0	1

Case-2 :

Switch $(S_1) \xrightarrow[(a \text{ to } c)]{} \text{OFF} \rightarrow \text{Logic-0}$

Switch $(S_2) \xrightarrow[(a \text{ to } b)]{} \text{ON} \rightarrow \text{Logic-1}$

Then circuit becomes as,



Now, it is clear that above circuit is not completed a loop and current I will not flow through the bulb and bulb goes OFF/logic-0.

Thus,

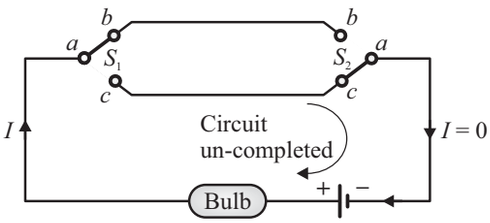
S_1	S_2	Bulb
0	1	0

Case-3 :

Switch $(S_1) \xrightarrow[(a \text{ to } b)]{} \text{ON} \rightarrow \text{Logic-1}$

Switch $(S_2) \xrightarrow[(a \text{ to } c)]{} \text{OFF} \rightarrow \text{Logic-0}$

Then circuit becomes as,



Now, it is clear that above circuit is not completed a loop and current I will not flow through the bulb and bulb goes OFF/logic-0.

Thus,

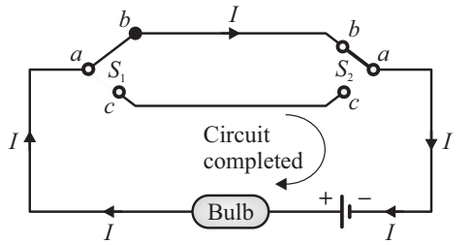
S_1	S_2	Bulb
1	0	0

Case-4 :

Switch $(S_1) \xrightarrow[(a \text{ to } b)]{} \text{ON} \rightarrow \text{Logic-1}$

Switch $(S_2) \xrightarrow[(a \text{ to } b)]{} \text{ON} \rightarrow \text{Logic-1}$

Then circuit becomes as,



Now, it is clear that above circuit is completed a loop and current I flow through the bulb and bulb goes ON/logic-1/glows.

Thus,

S_1	S_2	Bulb
1	1	1

So, we can combine all 4-cases together then we get complete truth table as,

Switch		Bulb
S_1	S_2	
0	0	1
0	1	0
1	0	0
1	1	1

Thus, above table satisfy the truth table of EX-NOR gate, so its expression becomes,

$\Rightarrow \quad S_1 \odot S_2$

$\Rightarrow \quad \overline{S_1 \oplus S_2}$

Hence, the correct option is (C).

Key Point

(i) 2-input EX-OR and EX-NOR logic gate are always complemented to each other.

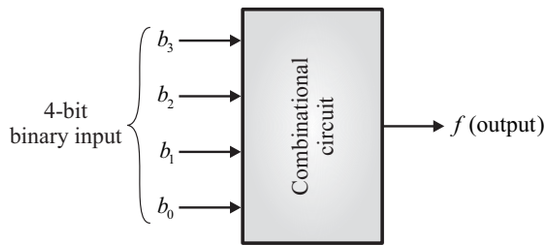
i.e. $A \oplus B = \overline{A \odot B}$

$A \odot B = \overline{A \oplus B}$

(ii) 3-input EX-OR and EX-NOR logical expression are always equal to each other.
i.e. $A \oplus B \oplus C = A \odot B \odot C$

7 (B)

According to question,



$$f \rightarrow \begin{cases} 1 \Rightarrow \text{If decimal value of } (b_3b_2b_1b_0) \geq 5 \\ 0 \Rightarrow \text{If decimal value of } (b_3b_2b_1b_0) < 5 \end{cases}$$

Decimal value of $(b_3b_2b_1b_0)$ is less between 0 to 9 only, after that all decimal value of $(b_3b_2b_1b_0)$ will treat as don't care (X).

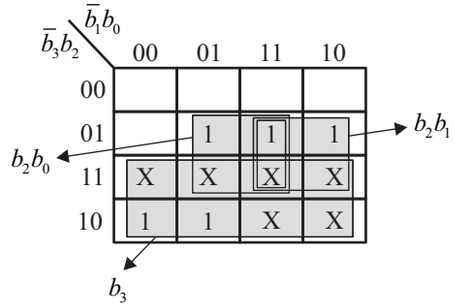
Thus, truth table according to given condition is:

Input				Output	Min terms
b_3	b_2	b_1	b_0	f	m
0	0	0	0	0	m_0
0	0	0	1	0	m_1
0	0	1	0	0	m_2
0	0	1	1	0	m_3
0	1	0	0	0	m_4
0	1	0	1	1	$\boxed{m_5}$
0	1	1	0	1	$\boxed{m_6}$
0	1	1	1	1	$\boxed{m_7}$
1	0	0	0	1	$\boxed{m_8}$
1	0	0	1	1	$\boxed{m_9}$
1	0	1	0	X	m_{10}
1	0	1	1	X	m_{11}
1	1	0	0	X	m_{12}
1	1	0	1	X	m_{13}
1	1	1	0	X	m_{14}
1	1	1	1	X	m_{16}

Thus, output (f) can be written as,

$$f = \Sigma m(5, 6, 7, 8, 9) + \Sigma d(10, 11, 12, 13, 14, 15)$$

So, output (f) can be minimized by K-map in SOP format as,

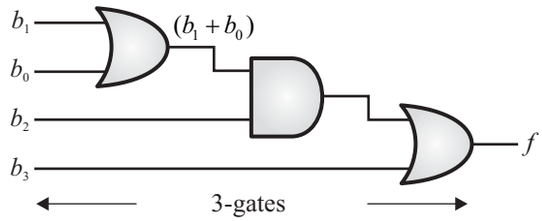


So, output (f) in minimized form as,

$$f = b_3 + b_2b_1 + b_2b_0$$

$$f = b_3 + b_2(b_1 + b_0)$$

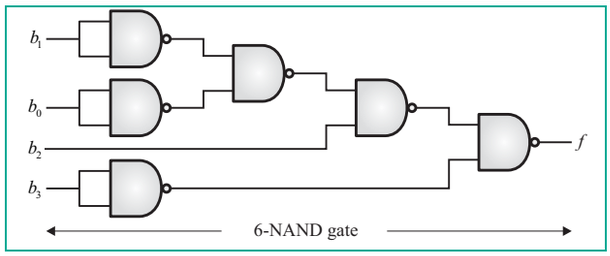
So, f can be implemented using only AND/OR/NOT gates as,



Thus, minimum number of 3-gates are required.

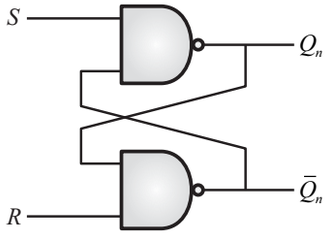
Hence, the correct option is (B).

Note : In the same question if examine asked, minimum number 2-input NAND gate required to implement the function $(f) = b_3 + b_2(b_1 + b_0)$ then answer should be 6 and its implementation shown below,



8 (D)

Given : Cross coupled *R-S* flip-flop using only two NAND gate is shown below,



(Assume initially $Q_n = 0$ and $\overline{Q}_n = 1$)

Here, question asked about oscillation means output of flip-flop will toggle between 1 and 0.

From option (A) :

(i) 1st apply $SR = 11$, then output $Q_n = 0$ and $\overline{Q}_n = 1$ (same as initial state).

(ii) 2nd time, apply $SR = 00$, then output $Q_n = 1$ and $\overline{Q}_n = 1$.

(iii) No more input in option (A) so continue with $SR = 00$ in 3rd time and so on, then again output equals to $Q_n = 1$ and $\overline{Q}_n = 1$.

Thus, output Q is,

$$0 \to 1 \to 1 \to .....$$

So, after 1st output, it will fix at logic 1.

Thus, the given input combination 11, 00 does not lead oscillation at output.

So, option (A) is incorrect.

From option (B) :

(i) First apply $SR = 01$, then output $Q_n = 1$ and $\overline{Q}_n = 0$.

(ii) Second time, apply $SR = 10$, then output $Q_n = 0$ and $\overline{Q}_n = 1$.

(iii) No more input in option (B) so continue with $SR = 10$ in 3rd time and so on, then again output equals to $Q_n = 0$ and $\overline{Q}_n = 1$.

Thus, output Q is,

$$1 \to 0 \to 0 \to .....$$

So, after 1st output, it will fix at logic 0.

Thus, the given input combination 01, 10 does not lead oscillation at output.

So, option (B) is incorrect.

From option (C) :

(i) First apply $SR = 10$, then output $Q_n = 0$ and $\overline{Q}_n = 1$.

(ii) Second time, apply $SR = 01$, then output $Q_n = 1$ and $\overline{Q}_n = 0$.

(iii) No more input in option (C) so continue with $SR = 01$ in 3rd time and so on, then again output equals to $Q_n = 1$ and $\overline{Q}_n = 0$.

Thus, output Q is,

$$0 \to 1 \to 1 \to .....$$

So, after 1st output, it will fix at logic 1.

Thus, the given input combination 10, 01 does not lead oscillation at output.

So, option (C) is incorrect.

From option (D) :

(i) First apply $SR = 00$, then output $Q_n = 1$ and $\overline{Q}_n = 1$.

(ii) Second time, apply $SR = 11$, then output $Q_n = 0$ and $\overline{Q}_n = 1$.

(iii) No more input in option (D) so continue with $SR = 11$ in 3rd time and so on, then again output will changed and it becomes $Q_n = 1$ and $\overline{Q}_n = 0$ and again

4^{th} time apply input $SR = 11$, then again output changed to $Q_n = 0$ and $\overline{Q}_n = 1$ and this will continue.

So, output Q becomes,

$$1 \to 0 \to 1 \to 0 .....$$

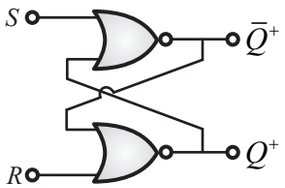
Thus, output continuously toggled between 1 and 0.

So, input combination (00, 11) leads oscillation in given flip-flop.

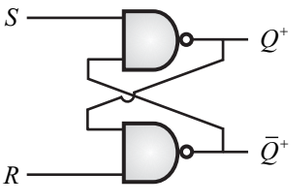
Hence, the correct option is (D).

Key Point

- In NOR latch if inputs are (0, 0), then outputs are previous state/hold [it may (0, 1) or (1, 0)].
- In NOR latch if inputs are (1, 1), then outputs are invalid state [it may (0, 0)].

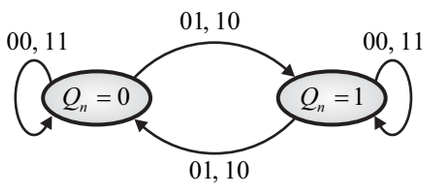


- In NAND latch if inputs are (1, 1), then outputs are previous states/hold [it may be (0, 1) or (1, 0)].
- In NAND latch if inputs are (0, 0), then outputs are invalid state [it may (1, 1)].



9 (D)

Given : state diagram is shown below,



Method 1

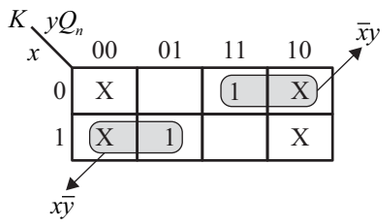
Assuming x, y are the inputs of given circuit and Q_n as previous state Q_n^+ as next state. So, state table can be formed from given state diagram is,

Inputs		Previous state	Next state
x	y	Q_n	Q_n^+
0	0	0	0
1	1	0	0
0	1	0	1
1	0	0	1
0	0	1	1
1	1	1	1
1	0	1	0
0	1	1	0

Now, according to excitation table of JK-flip-flop, JK inputs is decided as,

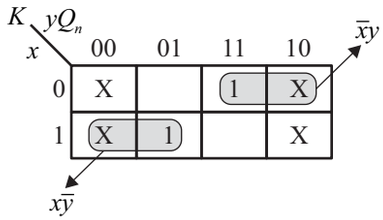
x	y	Q_n	Q_n^+	J	K
0	0	0	0	0	X
1	1	0	0	0	X
0	1	0	1	1	X
1	0	0	1	1	X
0	0	1	1	X	0
1	1	1	1	X	0
0	1	1	0	X	1
1	0	1	0	X	1

So, 3-variable (x, y, Q_n) K-map for J and K are,



$$\Rightarrow \quad J = x\overline{y} + \overline{x}y$$

$$\Rightarrow \quad J = x \oplus y$$



$\Rightarrow \quad K = x\bar{y} + \bar{x}y$

$\Rightarrow \quad K = x \oplus y$

Thus, minimized Boolean expression of J and K are $x \oplus y$ and $x \oplus y$ respectively.

Hence, the correct option is (D).

Method 2

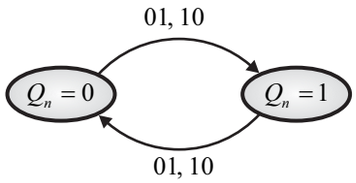
From state diagram it is clear that,

(i) If input $xy = 00$ or 11

Then machine will remain in previous state because of self-loop so,

Output (Q_n^+) = Previous state/Hold

(ii) If input $xy = 01$ or 10



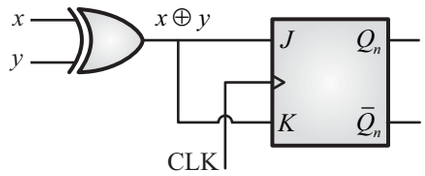
Then, machine will toggle between 0 to 1 or 1 to 0, so

Output (Q_n^+) = Toggle mode.

So, based on above two condition, we have to choose combinational circuit in such a way that,

- When $xy = 00$ or 11, then $J = K = 0$ so that output Q_n^+ is remain in previous state/hold.
- When $xy = 01$ or 10, then $J = K = 1$, so that output Q_n^+ is in toggle mode.

So, it is possible only with EX-OR gate with input x and y, so circuit becomes as



Thus, $J = K = x \oplus y$

Hence, the correct option is (D).

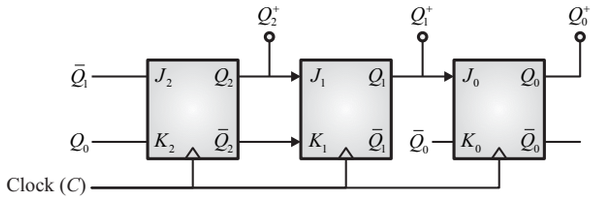
Key Point

(i) When both input of JK flip-flop are at logic-1 then output of JK flip-flop is in toggle state.

(ii) When both input of JK flip-flop are at logic-0 then output of JK flip-flop is in hold/previous state.

10 (C)

Given : Circuit redraw here is shown below,



Method 1

It is a 3-bit synchronous sequential circuit. So, inputs to each flip-flops are,

$$J_2 = \bar{Q}_1,\ J_1 = Q_2,\ J_0 = Q_1$$

$$K_2 = Q_0,\ K_1 = \bar{Q}_2,\ K_0 = \bar{Q}_0$$

Initially all flip-flop are cleared i.e.,

$$Q_2Q_1Q_0 = 000$$

So, state table can be formed as,

Previous state			Flip-flop input						CLK	Next state		
Q_2	Q_1	Q_0	$J_2(\bar{Q}_1)$	$K_2(Q_0)$	$J_1(Q_2)$	$K_1(\bar{Q}_2)$	$J_0(Q_1)$	$K_0(\bar{Q}_0)$		Q_2^+	Q_1^+	Q_0^+
-	-	-	-	-	-	-	-	-	0	0	0	0
0	0	0	1	0	0	1	0	1	1	1	0	0
1	0	0	1	0	1	0	0	1	2	1	1	0
1	1	0	0	0	1	0	1	1	3	1	1	1

Initially

So, state sequence for next 3-cycle is 100, 110, 111.

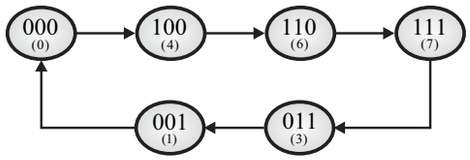
Hence, the correct option is (C).

Method 2

Given : circuit is 3-bit Johnson counter using JK-flip-flop, so its state table is shown below,

Clock	Q_2^+	Q_1^+	Q_0^+	Decimal value
0	0	0	0	0
1	1	0	0	4
2	1	1	0	6
3	1	1	1	7
4	0	1	1	3
5	0	0	1	1
6	0	0	0	0 (Repeated)

So, counting sequence is,



Here, question asked about, state sequence for next 3-cycle is $100 \rightarrow 110 \rightarrow 111$.

Hence, the correct option is (C).

❖❖❖❖

GATE ACADEMY®

steps to success...

Pen Drive & G-Drive Course

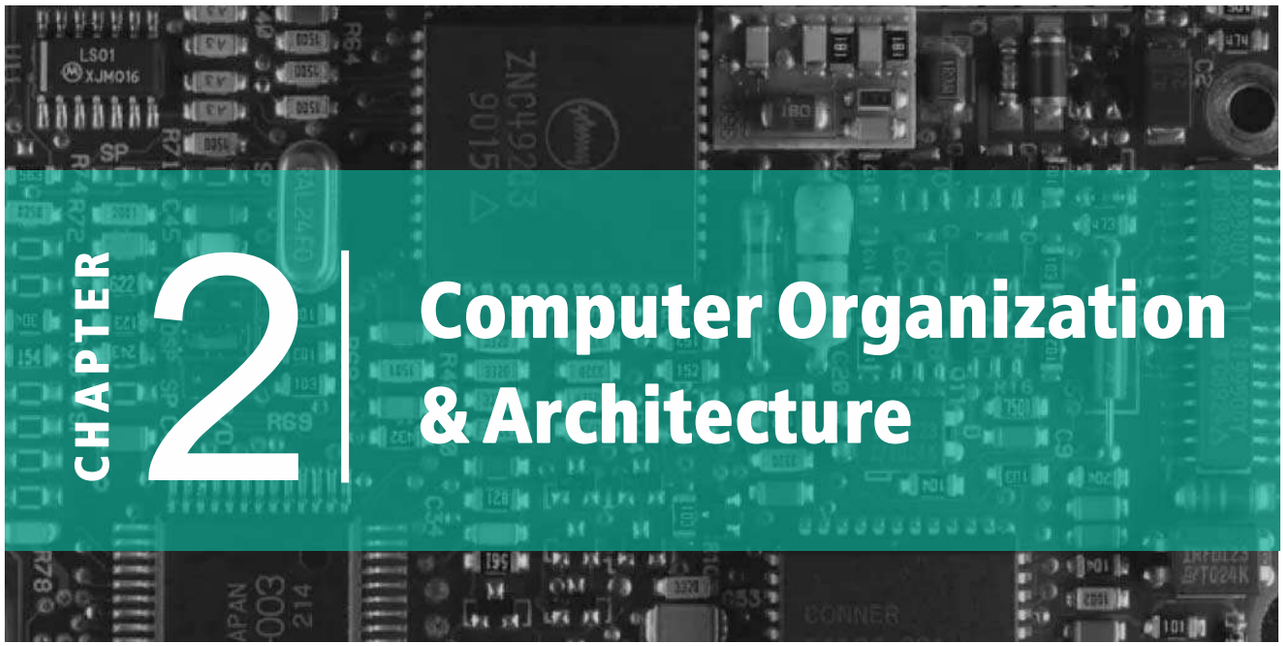
- It is best offline mode of video course.
- Dedicated discussion and doubt portal managed by our faculties.
- Complete packaged course with all reading content and Online Test Series.
- Ready to ship course, It will be shipped to you within 2 days of ordering.

Tablet Course

- Tablet course will be delivered through Samsung Galaxy Tab A (WiFi only).
- All GATE ACADEMY apps will come pre loaded.
- Tablet will be controlled by our team for the course duration.
- After course duration the tablet will be made open for your use.
- The Course content will be exactly same as VOD course but you do not need Internet connection to play the video (as it is an offline course

For Doubt and Discussion you need to download GATE ACADEMY learning app, Subject matter expert/ faculty will be there to solve your query

A DEDICATED TECH TEAM TO SOLVE TECH QUERIES AND HANDLE TECHNICAL ISSUES.



CHAPTER 2 Computer Organization & Architecture

Marks Distribution of Computer Organization & Architecture in Previous Year GATE Papers.

Exam Year	1 Mark Ques.	2 Marks Ques.	Total Marks
2003	2	3	8
2004*	1	7	15
2005*	4	8	20
2006*	1	7	15
2007*	2	6	14
2008*	-	12	24
2009	2	4	10
2010	1	4	9
2011	2	2	6
2012	1	4	9
2013	1	3	7
2014 Set-1	1	3	7
2014 Set-2	1	3	7
2014 Set-3	1	2	5

Exam Year	1 Mark Ques.	2 Mark Ques.	Total Marks
2015 Set-1	1	1	3
2015 Set-2	1	2	5
2015 Set-3	1	2	5
2016 Set-1	1	2	5
2016 Set-2	1	5	11
2017 Set-1	3	4	11
2017 Set-2	1	3	7
2018	3	3	9
2019	2	1	4
2020	5	3	11
2021 Set-1	2	4	10
2021 Set-2	2	3	8

*** CS and IT combined**

Syllabus : Computer Organization & Architecture

Machine instructions and addressing modes. ALU, data-path and control unit. Instruction pipelining, pipeline hazards. Memory hierarchy: cache, main memory and secondary storage; I/O interface (interrupt and DMA mode).

Contents : Computer Organization & Architecture

2003 IIT Madras

1 Consider the following assembly language program for a hypothetical processor. A, B, and C are 8-bit registers. The meanings of various instructions are shown as comments.

```
      MOV B, # 0      ;  B ← 0
      MOV C, # 8      ;  C ← 8
Z:    CMP C, # 0      ;  compare C with 0
      JZX             ;  jump to X if zero flag is set
      SUB C, # 1      ;  C ← C – 1
      RRC A, # 1      ;  right rotate A through carry by one bit. Thus: if the initial values of A and the carry flag are a7 ... a0 and c0 respectively, their values after the execution of this instruction will be c0a7 ... a1 and a0 respectively
      JC Y            ;  jump to Y if carry flag is set
      JMP Z           ;  jump to Z
Y:    ADD B, # 1      ;  B ← B + 1
      JMP Z           ;  jump to Z
X:
```

If the initial value of register A is A_0 the value of register B after the program execution will be

(A) the number of 0 bits in A_0

(B) the number of 1 bits in A_0

(C) A_0

(D) 8

2005 IIT Bombay

2 An instruction set of a processor has 125 signals which can be divided into 5 groups of mutually exclusive signals as follows:

Group 1 : 20 signals, Group 2 : 70 signals, Group 3 : 2 signals, Group 4 : 10 signals, Group 5 : 23 signals.

How many bits of the control words can be saved by using vertical microprogramming over horizontal microprogramming?

(A) 0 (B) 103

(C) 22 (D) 55

2008 IISc Bangalore

3 Consider a machine having a two-way set associative data cache of size 64 Kbytes and block size 16 bytes. The cache is managed using 32 bit virtual address and the page size is 4 Kbytes. A program to be run on this machine begins as follows:

```
Double ARR [1024] [1024]
inti, j;
/* Initialize array ARR to 0.0*/
for (i = 0; i < 1024; i++)
for ( j = 0; j < 1024; j++)
ARR [i][ j] = 0.0;
```

The size of double 8 bytes. Array ARR is located in memory starting at the beginning of virtual page $0 \times FF000$ and stored in row major order. The cache is initially empty and no pre-fetching is done. The only data memory references made by the program are those to array ARR….

Which of the following array elements has the same cache index as ARR [0] [0]?

(A) ARR [0] [4] (B) ARR [4] [0]

(C) ARR [0] [5] (D) ARR [5] [0]

2009 IIT Roorkee

4 Consider a 4-stage pipeline processor. The number of cycles needed by the four instructions I_1, I_2, I_3, I_4 in stages S_1, S_2, S_3, S_4 is shown below.

	S_1	S_2	S_3	S_4
I_1	2	1	1	1
I_2	1	3	2	2
I_3	2	1	1	3
I_4	1	2	2	2

What is the number of cycles needed to execute the following loop?

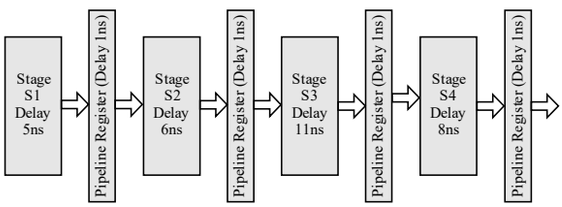
For (i = 1 to 2){(I_1; I_2; I_3; I_4)}

(A) 16 (B) 23

(C) 28 (D) 30

2011 IIT Madras

5 Consider an instruction pipeline with four stages (S1, S2, S3 and S4) each with combinational circuit only. The pipeline registers are required between each stage and at the end of the last stage. Delays for the stages and for the pipeline registers are as given in the figure.



What is the approximate speedup of the pipeline in steady state under ideal conditions when compared to the corresponding non-pipeline implementation?

(A) 4.0 (B) 2.5

(C) 1.1 (D) 3.0

6 On a non-pipelined sequential processor, a program segment, which is a part of the interrupt service routine, is given to transfer 500 bytes from an I/O device to memory.

Initialize the address register
Initialize the count to 500
LOOP : Load a byte from device
Store in memory at address
given by address register
Increment the address register
Decrement the count
If count ! = 0 go to LOOP

Assume that each statement in this program is equivalent to machine instruction which takes one clock cycle to execute if it is a non-load/store instruction. The load-store instructions take two clock cycles to execute.

The designer of the system also has an alternate approach of using DMA controller to implement the same transfer. The DMA controller requires 20 clock cycles for initialization and other overheads. Each DMA transfer cycle takes two clock cycles to transfer one byte of data from the device to the memory.

What is the approximate speedup when the DMA controller based design is used in place of the interrupt driven program based input-output?

(A) 3.4 (B) 4.4
(C) 5.1 (D) 6.7

2013 IIT Bombay

7 Consider an instruction pipeline with five stages without any branch prediction: Fetch Instruction (FI), Decode Instruction (DI), Fetch Operand (FO), Execute Instruction (EI) and Write Operand (WO). The stage delays for FI, DI, FO, EI and WO are 5 ns, 7ns, 10 ns, 8 ns and 6 ns, respectively. There are intermediate storage buffers after each stage and the delay of each buffer is 1 ns. A program consisting of 12 instructions $I_1, I_2, I_3, ..., I_{12}$ is executed in this pipelined processor. Instruction I_4 is the only branch instruction and its branch target is I_9. If the branch is taken during the execution of this program, the time (in ns) needed to complete the program is

(A) 132 (B) 165
(C) 176 (D) 328

2014 IIT Kharagpur

8 Consider the following processors (ns stands for nanoseconds). Assume that the pipeline registers have zero latency.

P_1 : Four-stage pipeline with stage latencies 1 ns, 2 ns, 2 ns, 1 ns.

P_2 : Four-stage pipeline with stage latencies 1 ns, 1.5 ns, 1.5 ns, 1.5 ns.

P_3 : Five-stage pipeline with stage latencies 0.5 ns, 1 ns, 1 ns, 0.6 ns, 1 ns.

P_4 : Five-stage pipeline with stage latencies 0.5 ns, 0.5 ns, 1 ns, 1 ns, 1.1 ns.

Which processor has the highest peak clock frequency?

(A) P1 (B) P2
(C) P3 (D) P4

2017 IIT Roorkee

9 The read access times and the hit ratios for different caches in a memory hierarchy are as given below.

Cache	Read access time (in nanoseconds)	Hit ratio
I-cache	2	0.8
D-cache	2	0.9
L2-cache	8	0.9

The read access time of main memory is 90 nanoseconds. Assume that the caches use the referred-words-first read policy and the write back policy. Assume that all the caches are direct mapped caches. Assume that the dirty bit is always 0 for all the blocks in the caches. In execution of a program, 60% of memory reads are for instruction fetch and 40% are for memory operand fetch. The average read access time in nanoseconds (up to 2 decimal places) is_____

2021 IIT Bombay

10 If the numerical value of a 2-byte unsigned integer on a little endian computer is 255 more than that on a big endian computer, which of the following choices represent(s) the unsigned integer on a little endian computer?

(A) 0x6665 (B) 0x0001
(C) 0x4243 (D) 0x0100

Answers Computer Organization & Architecture

1.	B	2.	B	3.	B	4.	D	5.	B
6.	A	7.	B	8.	C	9.	4.72	10.	A, D

Explanations Computer Organization & Architecture

1 (B)

Given :

```
      MOV B, # 0    ;  B ← 0
      MOV C, # 8    ;  C ← 8
Z:    CMP C, # 0    ;  compare C with 0
      JZX           ;  jump to X if zero flag is set
      SUB C, # 1    ;  C ← C – 1
      RRC A, # 1    ;  right rotate A through carry by one bit. Thus: if the initial values of A and the carry flag are a7 ... a0 and c0 respectively, their values after the execution of this instruction will be c0a7 ... a1 and a0 respectively
      JC Y          ;  jump to Y if carry flag is set
      JMP Z         ;  jump to Z
Y:    ADD B, # 1    ;  B ← B + 1
      JMP Z         ;  jump to Z
X:
```

And a hypothetical processor with A, B, and C as 8-bit registers.

Let $A_0 = 7$ (00000111)

Initially $B = 0, C = 8$

$\because C \neq 0;\ \therefore$ zeroflag $= 0;\ C = 7 \quad (\because C = C - 1)$

Rotate A : $A = 00000011, C_0 = 1; B = 1$

$(\because B = B + 1)$

Next iteration,

$\because C \neq 0;\ \therefore$ zeroflag $= 0;\ C = 6 \quad (\because C = C - 1)$

Rotate A : $A = 10000001, C_0 = 1; B = 2$

$(\because B = B + 1)$

Again

$\because C \neq 0;\ \therefore$ zeroflag $= 0;\ C = 5 \quad (\because C = C - 1)$

Rotate A : $A = 11000000, C_0 = 1; B = 3$

$(\because B = B + 1)$

For next 5 iterations (until C is 0) $C_0 = 0$ hence B remains same.

$\because \quad B = 3 =$ number of 1 bits in A_0

Hence, the correct option is (B).

2 (B)

Given :

Total number of signals = 125

In horizontal micro programming number of bits required = Number of signals

Therefore, number of bits = 125

Number of signals in group 1 = 20

Number of signals in group 2 = 70

Number of signals in group 3 = 2

Number of signals in group 4 = 10

Number of signals in group 5 = 23

Key Point

In vertical micro programming control signals are encoded.

So, no. of bits for,

Group 1 : $\log_2 20 = 5$ bits

Group 2 : $\log_2 70 = 7$ bits

Group 3 : $\log_2 2 = 1$ bits

Group 4 : $\log_2 10 = 4$ bits

Group 5 : $\log_2 23 = 5$ bits

Total = 22 bits

Number of bits saved = Number of bits in horizontal – Number of bits in vertical micro programming $= 125 - 22 = 103$

Hence, the correct option is (B).

3 (B)

Given :

Array is ARR [1024][1024]

$$\begin{bmatrix} ARR[0][0], ARR[0][1], \ldots\ldots\ldots\ldots, ARR[0][1023] \\ ARR[1][0], ARR[1][1], \ldots\ldots\ldots\ldots, ARR[1][1023] \\ \vdots \\ ARR[1023][0], ARR[1023][1], \ldots\ldots\ldots ARR[1023][1023] \end{bmatrix}$$

1 element size = 8 B

Block size = 16 B

One block holds 2 elements

One row size = 1024 element

Number of blocks required to store one row = 512 blocks

Sets :

0	ARR [0][0], ARR [0][1]	ARR [4][0], ARR [4][1]
1	ARR [0][2], ARR [0][3]	ARR [4][2], ARR [4][3]
2	ARR [0][4], ARR [0][5]	ARR [4][4], ARR [4][5]
	1st row	5th row
511	ARR [0][1022], ARR [0][1023]	ARR [4][1022], ARR [4][1023]
512	ARR [1][0], ARR [1][1]	ARR [5][0], ARR [5][1]
	ARR [1][2], ARR [1][3]	
	2nd row	6th row
1023	ARR [1][1022], ARR [1][1023]	
1024	3rd row	7th row
1535		
1536		
	4th row	8th row
2047		

ARR [0][0] has the same cache index as ARR [4][0]

All the consecutive blocks will be stored in consecutive sets

B_0 block containing ARR[0][0], ARR[0][1] will be there in set 0

B_1, containing ARR[0][2], ARR[0][3] in set 1

B_2 containing ARR[0][4], ARR[0][5] in set 2

In this way, last block of the row

B_{511} containing ARR[0][1022], ARR[0][1023] will be in set 511 then next 3 rows will be stored in similar manner in subsequent sets till set 2047 (last set) containing last block of 4th row B_{2047} contains element ARR[3][1022] and ARR[3][1023]

Therefore next block belonging to 5th row goes to set 0 contains element

ARR[4][0], ARR[4][1]

Hence, the correct option is (B).

4 (D)

Given : Number of stages = 4

Stage delays :

	S_1	S_2	S_3	S_4
I_1	2	1	1	1
I_2	1	3	2	2
I_3	2	1	1	3
I_4	1	2	2	2

For 1 iteration the space time diagram will be

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_1	S_1	S_1	S_2	S_3	S_4										
I_2			S_1	S_2	S_1	S_2	S_3	S_3	S_4	S_4					
I_3				S_1	S_2	S_2	S_3	S_4	S_4		S_4				
I_4					--	--	S_1	--				S_4			

Therefore, 15 cycle is required for 1 iteration so, to execute the loop for 2 iteration 30 cycles will be required.

Hence, the correct option is (D).

Note

If in the question loop level parallelism would have given, then space-time diagram :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
I_1	S_1	S_1	S_1		--	--		--	S_1	S_1	S_1	--	--		--	--	S_1	--	S_2	S_2	S_3	S_3	--	S_4	S_4
I_2			S_2	S_1	S_1	--	S_1	--	S_2	S_2	S_2	S_3	--	S_1	--	--	S_2	S_3	S_3	S_4	S_4	S_4	S_4	S_4	S_4	S_4
I_3				S_2	S_2	S_2	S_2	--	S_3	--	S_3	S_4	S_4	S_4	S_4	S_4	S_3	--	S_4	S_3	--	S_4	S_4	S_4		
I_4					S_3	S_3	S_3	S_3	S_4	S_4	S_4	S_4	S_4		S_4											

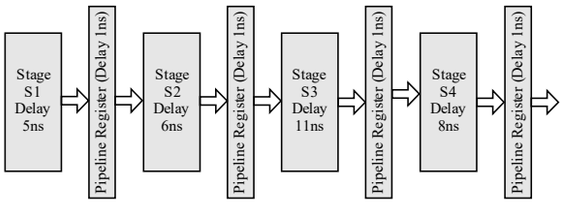
Number of cycles = 25

5 (B)

Given :

Number of stages = 4

Stage and Buffer delays :



Key Point

Execution time for pipeline =Stage delay + Register delay

Execution time for non pipe line

$$= 5+6+11+8 = 30 \text{ ns}$$

Execution time for pipeline =Stage delay + Register delay

$$= 11+1 = 12 \text{ ns}$$

$$\text{Speed up} = \frac{30}{12} = 2.5 \text{ ns}$$

$$\left\{\lim_{n\to\infty} \frac{30n}{36+12n} = \frac{30}{12}\right\}$$

Hence, the correct answer is (B).

6 (A)

Given :

Number of clock cycle for each non-load/store instructions = 1

Number of clock cycles for each load/store instructions = 2

		Cycles
	Initialize the address register	1
	Initialize the count to 500	1
LOOP :	Load a byte from device	2
	Store in memory at address	2
	given by address register	1
	Increment the address register	1
	Decrement the count	1
	If count ! = 0 go to LOOP	

For DMA approach

Number of clock cycles for initialization = 20

Number of clock cycles to transfer each byte = 2

Interrupt driven transfer time

$$= 1+1+500(2+2+1+1+1) = 3502$$

DMA based transfer time $= 20+500\times 2 = 1020$

$$\text{Speed up} = \frac{3502}{1020} \approx 3.4$$

Hence, the correct option is (A).

7 (B)

Given :

Number of stages = 5

Stage delays = 5, 7, 10, 8, 6 ns

Buffer delay = 1 ns

Number of instructions = 12

Method 1

According to the question branch is taken that means after I4, I9 will be executed and further I10, I11, I12.

Space-time diagram :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	FI	DI	FO	EI	WO										
12		FI	DI	FI	EI	WO									
13			FI	DI	PO	EI	WO								
14				FI	DI	FO	EI	WO							
19					Stall	Stall	Stall	FI	DI	FO	EI	WO			
110									FI	DI	FO	EI	WO		
111										FI	DI	FO	EI	WO	
112											FI	DI	FO	EI	WO

Number of cycles required = 15 cycles

Execution time = No. of cycles × Cycle time

Stage delay = max(5, 7, 10, 8, 6))

Total delay = 10 + 1 = 11 ns

Execution time = 15 × 11 = 165 ns

Hence, the correct option (B).

Method 2

According to the question branch is taken that means after I4, I9 will be executed and further I10, I11, I12.

Therefore,

Number of instructions executed = 8

Number of cycles without hazards $= k + n - 1$

$= 5 + 8 - 1 = 12$ cycles

Number of stalls cycles because of branch instructions $= 4 - 1 = 3$

Total number of cycles $= 12 + 3 = 15$

Execution time $= 15 \times 11 = 165$ ns

Hence, the correct option (B).

8 (C)

Given :

Processor configurations :

P_1 : Four-stage pipeline with stage latencies 1 ns, 2 ns, 2 ns, 1 ns.

P_2 : Four-stage pipeline with stage latencies 1 ns, 1.5 ns, 1.5 ns, 1.5 ns.

P_3 : Five-stage pipeline with stage latencies 0.5 ns, 1 ns, 1 ns, 0.6 ns, 1 ns.

P_4 : Five-stage pipeline with stage latencies 0.5 ns, 0.5 ns, 1 ns, 1 ns, 1.1 ns.

Method 1

CT for $P_1 = \frac{1}{2} = 0.5$ GHz

CT for $P_2 = \frac{1}{1.5} = 0.67$ GHz

CT for $P_3 = \frac{1}{1} = 1$ GHz

CT for $P_4 = \frac{1}{1.1} = 0.90$ GHz

Hence, the correct option is (C).

Method 2

Find maximum stage delay of each process :

For P_1, maximum stage delay = 2 ns

For P_2, maximum stage delay = 1.5 ns

For P_3, maximum stage delay = 1 ns

For P_4, maximum stage delay = 1.1 ns

Since, clock cycle time $\propto \frac{1}{\max \text{ (stage delays)}}$

Processor with minimum stage delay will have highest frequency.

Therefore, P_3 has the highest clock frequency.

Hence, the correct option is (C).

Key Point

Clock cycle time $\propto \frac{1}{\max \text{ (stage delays)}}$

9 4.72

Given :

Read access time of cache :

Cache	Read access time (in nanoseconds)	Hit ratio
I-cache	2	0.8
D-cache	2	0.9
L2-cache	8	0.9

Read access time of memory = 90 ns

60% of memory reads are for instruction fetch and 40% are for memory operand fetch.

Let, H_I = hit ratio of I-cache

H_D = hit ratio of D-cache

H_{L2} = hit ratio of L2-cache

H_M = hit ratio of memory

T_I = access time of I-cache

T_D = access time of D-cache

T_{L2} = access time of L2-cache

T_M = access time of memory

Average instruction fetch time

$$= H_I T_I + (1-H_I)H_{L2}(T_I + T_{L2}) + (1-H_I)(1-H_{L2})H_M(T_I + T_{L2} + T_M)$$
$$= 0.8\times 2 + 0.2\times 0.9\times(8+2) + 0.2\times 0.1\times(90+8+2)$$
$$= 5.4 \text{ ns}$$

T_{avg} read = frequency × read cycle time

$$= 0.6\times 5.4 = 3.24 \text{ ns}$$

Average data read time

$$= H_D T_D + (1-H_D)H_{L2}(T_D + T_{L2}) + (1-H_D)(1-H_{L2})H_M(T_D + T_{L2} + T_M)$$
$$= 0.9\times 2 + 0.1\times 0.9\times(8+2) + 0.1\times 0.1\times(90+8+2)$$
$$= 3.7 \text{ ns}$$

T_{avg} read = frequency × read cycle time

$$= 0.4\times 3.7 = 1.48 \text{ ns}$$

Total time $= 3.24 \text{ ns} + 1.48 \text{ ns} = 4.72 \text{ ns}$

Hence, the correct answer 4.72.

10 (A, D)

Consider each option one by one :

Option (A) : 0x6665

On little endian (LE) : 0x6665

On Big endian (BE) : 0x6566

Converting LE into decimal :

$$(6665)_{16} = (26213)_{10}$$

Converting BE into decimal :

$$(6566)_{16} = (25958)_{10}$$

Subtracting BE from LE,

$$26213 - 25958 = 255$$

Clearly, LE = 255 + BE

Option (B) : 0x0001

On little endian (LE) : 0x0001

On Big endian (BE) : 0x0100

Converting LE into decimal :

$$(0001)_{16} = (1)_{10}$$

Converting BE into decimal :

$$(0100)_{16} = (256)_{10}$$

Subtracting BE from LE,

$$1 - 256 = -255$$

Option (C) : 0x4243

On little endian (LE) : 0x4243

On Big endian (BE) : 0x4342

Converting LE into decimal :

$$(4243)_{16} = (16963)_{10}$$

Converting BE into decimal :

$$(4342)_{16} = (17218)_{10}$$

Subtracting BE from LE,

$$16963 - 17218 = -255$$

Option (D) : 0x0100

On little endian (LE) : 0x0100

On Big endian (BE) : 0x0001

Converting LE into decimal :

$$(0100)_{16} = (256)_{10}$$

Converting BE into decimal :

$$(0001)_{16} = (1)_{10}$$

Subtracting BE from LE,

$$256 - 1 = 255$$

Clearly, LE = 255 + BE

Hence, the correct options are (A), (D).

Key Point

Little Endian – In this scheme, low-order byte is stored on the starting address (A) and high-order byte is stored on the next address (A + 1).

Big Endian − In this scheme, high-order byte is stored on the starting address (A) and low-order byte is stored on the next address (A + 1).

Example :

Assume number 0x2568

LE representation of 0x2568 is :

$$\underbrace{25}_{\text{higher byte}}\ \underbrace{68}_{\text{lower byte}}$$

BE representation of 0x2568 is :

$$\underbrace{68}_{\text{higher byte}}\ \underbrace{25}_{\text{lower byte}}$$

❖❖❖❖

GATE ACADEMY®

steps to success...

Video on Demand

- VoD is the best online tool for self preparation students.
- Available for Android and Windows.
- All you need is decent internet connection and our GATE ACADEMY Learning App.
- All reading content will be provided within the App only.
- Dedicated discussion and doubt portal managed by our faculties.

Live Classroom Program

GATE LIVE

- Online Live Classroom program by team GATE Academy, the exact feel of offline Classroom by being in your comfort zone.
- Available only on Android app (Gate Academy Live)
- You can also watch it after live class as video on demand within your given and limited watch time.
- You can ask anything, a doubt clearing session with chat box during the class.
- All reading content will be provided within the live app only.

For Doubt and Discussion you need to download GATE ACADEMY learning app, Subject matter expert/ faculty will be there to solve your query

A DEDICATED TECH TEAM TO SOLVE TECH QUERIES AND HANDLE TECHNICAL ISSUES.

CHAPTER 3 Operating System

Marks Distribution of Operating System in Previous Year GATE Papers.

Exam Year	1 Mark Ques.	2 Marks Ques.	Total Marks
2003	2	5	12
2004*	3	4	11
2005*	-	2	4
2006*	1	8	17
2007*	2	6	14
2008*	2	5	12
2009	2	5	12
2010	3	2	7
2011	4	2	8
2012	1	4	9
2013	2	4	10
2014 Set-1	2	3	8
2014 Set-2	1	3	7
2014 Set-3	1	3	7

Exam Year	1 Mark Ques.	2 Mark Ques.	Total Marks
2015 Set-1	2	4	10
2015 Set-2	2	3	8
2015 Set-3	2	2	6
2016 Set-1	1	4	9
2016 Set-2	1	3	7
2017 Set-1	2	2	6
2017 Set-2	2	2	6
2018	2	3	8
2019	2	4	10
2020	2	4	10
2021 Set-1	3	2	7
2021 Set-2	2	3	8

*** CS and IT combined**

Syllabus : Operating System

System calls, processes, threads, inter-process communication, concurrency and synchronization. Deadlock. CPU and I/O scheduling. Memory management and virtual memory. File systems.

Contents : Operating System

1996 IISc Bangalore

1 A 1000 Kbyte memory is managed using variable partitions but no compaction. It currently has two partitions of size 200 Kbytes and 260 Kbytes respectively. The smallest allocation request in Kbytes that could be denied is for

(A) 151 (B) 181
(C) 231 (D) 541

1999 IIT Bombay

2 Which of the following is/are advantages of virtual memory?

(A) Faster access to memory on an average.

(B) Processes can be given protected address spaces.

(C) Linker can assign addresses independent of where the program will be loaded in physical memory.

(D) Programs larger than the physical memory size can be run.

2004 IIT Delhi

3 A disk has 200 tracks (numbered 0 through 199). At a given time, it was servicing the request of reading data from track 120, and at the previous request, service was for track 90. The pending requests (in order of their arrival) are for track numbers.

30 70 115 130 110 80 20 25.

How many times will the head change its direction for the disk scheduling policies SSTF(Shortest Seek Time First) and FCFS (First Come Fist Serve)

(A) 2 and 3 (B) 3 and 3
(C) 3 and 4 (D) 4 and 4

2007 IIT Kanpur

4 Consider n jobs $J_1, J_2, ...J_n$ such that job J_i has execution time t_i and a non-negative integer weight w_i. The weighted mean completion time of the jobs is defined to be $\frac{\Sigma_{i=1}^{n} w_i T_i}{\Sigma_{i=1}^{n} w_i}$, where T_i is the completion time of job J_i. Assuming that there is only one processor available, in what order must the jobs be executed in order to minimize the weighted mean completion time of the jobs?

(A) Non-decreasing order of t_i

(B) Non-increasing order of w_i

(C) Non-increasing order of $w_i t_i$

(D) Non-increasing order of w_i / t_i

2009 IIT Roorkee

5 Consider a system with 4 types of resources R1 (3 units), R2 (2 units), R3 (3 units), R4 (2 units). A non-preemptive resource allocation policy is used. At any given instance, a request is not entertained if it cannot be completely satisfied. Three process P1, P2, P3 request the resources as follows if executed independently.

Process P1 :
t = 0: requests 2 units of R2
t = 1: requests 1 unit of R3
t = 3: request 2 units of R1
t = 5: releases 1 unit of R2 and 1 unit of R1
t = 7: releases 1 unit of R3
t = 8: requests 2 units of R4
t = 10: finishes

Process P2 :

t = 0: requests 2 units of R3

t = 2: requests 1 unit of R4

t = 4: request 1 unit of R1

t = 6: releases 1 unit of R3

t = 8: finishes

Process P3 :

t = 0: requests 1 unit of R4

t = 2: requests 2 units of R1

t = 5: releases 2 units of R1

t = 7: requests 1 unit of R2

t = 8: requests 1 unit of R3

t = 9: finishes

Which one of the following statements is **TRUE** if all three processes run concurrently starting at time t = 0?

(A) All processes will finish without any deadlock

(B) Only P1 and P2 will be in deadlock

(C) Only P1 and P3 will be in deadlock

(D) All three processes will be in deadlock

2013 IIT Bombay

6 A certain computation generates two arrays a and b such that $a[i] = f(i)$ for $0 \le i < n$ and $b[i] = g(a[i])$ for $0 \le i < n$. Suppose this computation is decomposed into two concurrent processes X and Y such that X computes the array a and Y computes the array b. The processes employ two binary semaphores R and S, both initialized to zero. The array a is shared by the processes are shown below.

Process X :

```
private i;
for(i=0;i<n; i++)
{
    a[i] = f(i);
    Exit X (R,S);
}
```

Process Y :

```
private i;
for(i = 0; i<n; i++)
{   Entry Y (R,S);
    b[i] = g(a[i]);
}
```

Which one of the following represents the CORRECT implementation of Exit X and Entry Y?

(A)
```
Exit X(R, S)
{
  P (R);
  V (S);
}
Entry Y (R, S)
{
  P (S);
  V (R);
}
```

(B)
```
Exit X(R, S)
{
  V (R);
  V (S);
}
Entry Y (R, S)
{
P (R);
P (S);
}
```

(C)
```
Exit X (R, S)
{
  P (S);
  V (R);
}
Entry Y (R, S)
{
  V (S);
  P (R);
}
```

(D)
```
Exit X (R,S)
{
  V (R);
  P (S);
}
Entry Y (R, S)
{
  V (R);
  P (S);
}
```

2015 IIT Kanpur

7 For the processes listed in the following table, which of the following scheduling schemes will give the lowest turnaround time?

Process	Arrival time	Processing time
A	0	3
B	1	6
C	4	4
D	6	2

(A) First Come First Serve

(B) Non- preemptive Shortest Job First

(C) Shortest Remaining Time

(D) Round Robin with Quantum value two

8 Suppose the following disk request sequence (track number) for a disk with 100 tracks is given: 45, 20, 90, 10, 50, 60, 80, 25, 70. Assume that the initial position of the R/W head is on track 50. The additional distance that will be traversed by the R/W head when the Shortest Seek Time First (SSTF) algorithm is used compared to the SCAN (Elevator) algorithm (assuming that SCAN algorithm moves towards 100 when it starts execution) is _____ tracks.

2018 IIT Guwahati

9 In a system, there are three types of resources: E, F and G. Four processes P_0, P_1, P_2 and P_3 execute concurrently. At the outset, the processes have declared their maximum resource requirements using a matrix named Max as given below. For example, Max $[P_2, F]$ is the maximum number of instances of F that P_2 would require. The number of instances of the resources allocated to the various processes at any given state is given by a matrix named Allocation. Consider a state of the system with the Allocation matrix as shown below, and in which 3 instances of E and 3 instances of F are the only resources available.

Allocation	E	F	G
P_0	1	0	1
P_1	1	1	2
P_2	1	0	3
P_3	2	0	0

Max	E	F	G
P_0	4	3	1
P_1	2	1	4
P_2	1	3	3
P_3	5	4	1

From the perspective of deadlock avoidance, which one of the following is true?

(A) The system is in *safe* state.

(B) The system is not in *safe* state, but would be *safe* if one more instance of E were available

(C) The system is not in *safe* state, but would be *safe* if one more instance of F were available

(D) The system is not in *safe* state, but would be *safe* if one more instance of G were available.

Answers Operating System

1	B	2.	C	3.	C	4.	D	5.	A
6.	B	7.	C	8.	10	9.	A		

Explanations Operating System

1 (B)

Given :

1000 KB

Initially whole memory is a hole, i.e. free memory, then a process comes and it is allocated exactly the needed space, which creates a partition. In this question we can safely assume that the two given partitions of 200KB and 260KB are allocated to two processes, because otherwise it will all be one single hole of 1000KB and the answer to the question would be 1001KB)

Now let's see all the possible cases:

CASE 1.

If 200KB and 260KB are contiguous, from the beginning of the memory (obviously after the OS partition), then hole has a size of 540KB, so any process asking up to 540KB can be allocated.

200 KB	260 KB	540 KB

CASE 2.

If 200KB and 260KB are contiguous but somewhere in the middle of the memory, then we have two holes left. Hole1 and Hole2 (Refer the image below). If the size of these holes is not equal, say Hole1 = 300KB and Hole2 = 240KB then processes asking for memory up to 300KB can be allocated.

	200 KB	260 KB	

But if we keep hole sizes equal, we can see that Hole 1 = Hole 2 = 270KB, thus maximum 270 KB only can be allocated.

CASE 3.

Arguing in a similar manner as in case 2, we can see that size of all three holes must be equal (i.e. 180KB), otherwise we'll have a hole which will allow a process needing more than 180KB to get allocated.

180 KB	200 KB	180 KB	260 KB	180 KB

So, 181 KB is the smallest allocation request that can be denied.

Hence, the correct option is (B).

Key Point

Variable Partitioning :

(i) The partitions are of variable length and number, AND

(ii) When a process is brought into main memory, it is allocated exactly as much memory as it requires and no more.

In **compaction** we realign the scattered holes to one end of memory, so that a larger hole can be created.

2 (C)

Virtual memory provides an interface through which process access the physical memory. So,

Option (A) : Faster access to memory on an average.

Swapping of pages increases time while virtual memory concept is there and it slower the execution as compared to direct access from physical memory. So, it is false.

Option (B) : Processes can be given protected address spaces.

Without virtual memory, it is difficult to give protected address space to processes as they will be accessing physical memory directly.

So, it is true.

Option (C) : Linker can assign addresses independent of where the program will be loaded in physical memory.

Some other method can be used to assign addresses independent of where the program.

So it is false.

Option (D) : Programs larger than the physical memory size can be run.

Virtual memory allows a process to run using a virtual address space which is larger in size then the address of physical memory and pages are swapped in between physical memory and virtual memory when physical memory gets fall. So, it is true.

Hence, the correct option is (C).

3 (C)

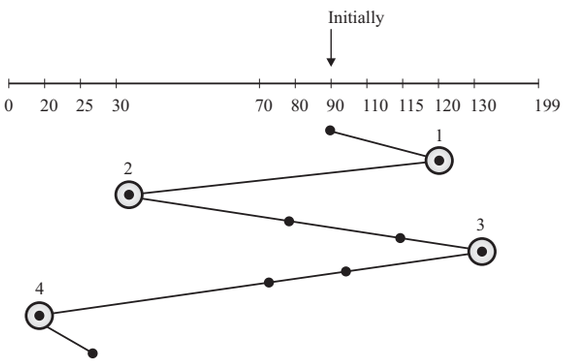
Given :

Number of tracks = 200

Pending requests :

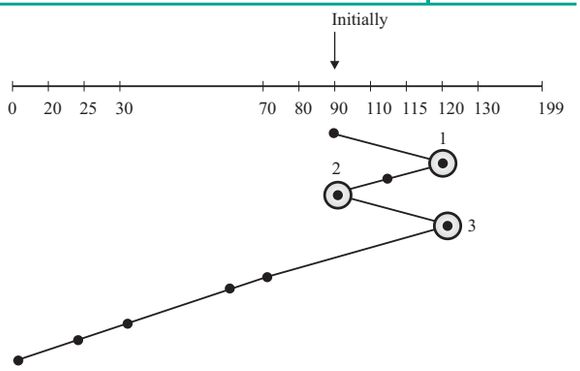
30, 70, 115, 130, 110, 80, 20, 25

Using FCFS :



So, 4 head direction change.

Using SRTF :



So, 3 head direction changes.

Hence, the correct option is (C).

4 (D)

Given :

Let's take an example :

Process	Weight	Execution time
P_1	1	3
P_2	2	5
P_3	3	2
P_4	4	4

For option (A) non decreasing t_i

$$= \frac{(3\times2+1\times5+4\times9+2\times14)}{10}$$

$$= \frac{(6+5+36+28)}{10}$$

$$= 7.5$$

For option (B) non-increasing w_i

$$= \frac{(4\times4+3\times6+2\times11+1\times14)}{10}$$

$$= \frac{(16+18+22+14)}{10}$$

$$= 7$$

For option (C) non-increasing w_it_i

$$= \frac{(16+2\times9+3\times11+1\times14)}{10}$$

$$= \frac{(16+18+33+14)}{10}$$

$$= 8.1$$

For option (D) non-increasing w_i/t_i

$$= \frac{(3\times2+4\times6+2\times11+1\times14)}{10}$$

$$= \frac{(6+10+22+14)}{10}$$

$$= 6.6$$

Minimum weighted mean obtained from non-increasing w_i / t_i.

Hence the correct option is (D).

5 (A)

Given :

Process P1 :

t = 0: requests 2 units of R2
t = 1: requests 1 unit of R3
t = 3: request 2 units of R1
t = 5: releases 1 unit of R2 and 1 unit of R1
t = 7: releases 1 unit of R3
t = 8: requests 2 units of R4
t = 10: finishes

Process P2 :

t = 0: requests 2 units of R3
t = 2: requests 1 unit of R4
t = 4: request 1 unit of R1
t = 6: releases 1 unit of R3
t = 8: finishes

Process P3 :

t = 0: requests 1 unit of R4
t = 2: requests 2 units of R1
t = 5: releases 2 units of R1
t = 7: requests 1 unit of R2
t = 8: requests 1 unit of R3
t = 9: finishes

At $t=0$

$\rightarrow P_1$ requests 2 units of R_2 which is granted.

$\rightarrow P_2$ requests 2 units of R_3 which is granted.

$\rightarrow P_3$ requests 1 units of R_4 which is granted.

Now available resources are,

R_1	R_2	R_3	R_4
3	0	1	1

At $t=1$,

$\rightarrow P_1$ requests 1 unit of R_3 which is granted because it is available.

Now available resources are

R_1	R_2	R_3	R_4
3	0	0	1

At $t=2$,

$\rightarrow P_2$ request 1 unit of R_4 which is granted.

$\rightarrow P_3$ requests 2 unit of R_1.

Now available resource are.

R_1	R_2	R_3	R_4
1	0	0	0

At $t=3$,

$\rightarrow P_1$ request 2 units of R_1 which cannot be granted and will wait for other processes to release.

Available resources are

R_1	R_2	R_3	R_4
1	0	0	0

At $t=4$,

$\rightarrow P_2$ requests 1 unit of R_1, which is granted.

Available resources are

R_1	R_2	R_3	R_4
1	0	0	0

At $t=5$,

$\rightarrow P_3$ releases 2 unit of R_1

Now Available resources are,

R_1	R_2	R_3	R_4
2	0	0	0

$\rightarrow$ Now P_1 releases 1 unit of R_2 and 1 unit of R_1. Now Available resources are

R_1	R_2	R_3	R_4
1	1	0	0

At $t=6$

$\rightarrow$ Now P_2 resources 1 unit of R_3

Now Available resources are.

R_1	R_2	R_3	R_4
1	1	1	0

At $t = 7$

$\rightarrow P_3$ requests 1 unit of R_2, Which is granted.

$\rightarrow P_1$ releases unit of R_3.

Now Available resources are.

R_1	R_2	R_3	R_4
1	0	2	0

At $t = 8$,

$\rightarrow P_2$ finishes and releases remaining resources.

So now Available resources.

R_1	R_2	R_3	R_4
2	0	3	1

$\rightarrow P_3$ requests 1 unit of R_2 which is granted.

Now Available resources are

R_1	R_2	R_3	R_4
2	0	1	1

$\rightarrow P_1$ requests 2 unit of R_4 which cannot be granted. So it will wait for other process to release them.

At $t = 9$

$\rightarrow P_3$ finishes, and releases rest of the resources. Now Available resources are

R_1	R_2	R_3	R_4
2	1	2	2

$\rightarrow$Now, P_1 can be granted with resources 2 units of R_4 for which it was waiting for.

Now Available resources are,

R_1	R_2	R_3	R_4
2	1	2	0

At $t = 10$

$\rightarrow P_1$ finishes its execution.

So, finally we can conclude that all finish without any deadlock.

Hence, the correct option is (A).

6 (B)

Given : code -

Process X :

```
private i;
for(i=0;i<n; i++)
{
    a[i] = f(i);
    Exit X (R,S);
}
```

Process Y :

```
private i;
for(i = 0; i<n; i++)
{   Entry Y (R,S);
    b[i] = g(a[i]);
}
```

The solution is using a binary semaphore. X and Y are two different processes whatever the values inserted by the processes in the array that will be used by processes Y after wards.

Option (A) :

```
Exit X(R, S)
 {
    P (R);
    V (S);
 }
Entry Y (R, S)
{
   P (S);
   V (R);
}
```

Suppose process X executes exit X then it will wait for R, and then process Y executes entry Y then it will wait for S. Since initially both binary semaphores are '0' no one will increment it both the processes will stuck in deadlock.

$\therefore$ Option (A) is wrong.

Option (B) :

```
Exit X(R, S)
 {
    V (R);
    V (S);
```

```
}
Entry Y (R, S)
{
    P (R);
    P (S);
}
```

It is wrong, because process X can insert multiple values in array and terminate but process Y can consume only one value.

∴ Option (B) is wrong.

Option (C) :

```
Exit X (R, S)
{
    P (S);
    V (R);
}
Entry Y (R, S)
{
    V (S);
    P (R);
}
```

Here take any sequence of operation of process X and process Y. First process X will wait for S which is incremented by process X.

There is no sequence of operation in which the value of R or S overlaps.

Hence both processes execute one after another.

∴ Option (C) is wrong.

Option (D) :

```
Exit X (R,S)
{
    V (R);
    P (S);
}
Entry Y (R, S)
{
    V (R);
    P (S);
}
```

Suppose first process X executes. It sets $R=1$ and then waits for S. Now after that process Y executes. It first sets $S=1$ and then decrement $R=0$. It comes again and then again sets $S=1$ (i.e., it overlaps the value of S) and then wait for R. Clearly here we lost one iteration of process X due to overlapping of value of S, and after ($n-1$) iteration process well stock.

∴ Option (D) is wrong.

Hence, the correct option is (B).

7 (C)

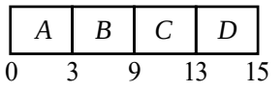
Given :

Process	Arrival time	Processing time
A	0	3
B	1	6
C	4	4
D	6	2

Key Point

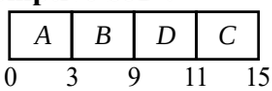
Turnaround Time = Completion Time – Arrival Time

(1) FCFS :



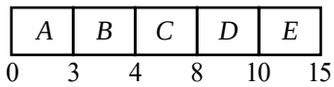
$$\text{Avg TAT} = \frac{3+8+9+9}{4} = \frac{29}{4} = 7.25$$

(2) Non preemptive SJF :



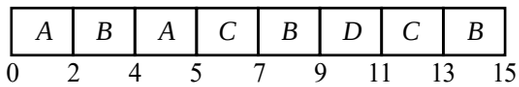
$$\text{Avg TAT} = \frac{3+8+5+11}{4} = 6.75$$

(3) SRTF :



$$\text{Avg TAT} = \frac{3+14+4+4}{4} = \frac{25}{4} = 6.25$$

(4) RR :



$$\text{Avg TAT} = \frac{5+14+9+5}{4} = \frac{33}{4} = 8.25$$

SRTF has lowest turn around time.

Hence, the correct option is (C).

8 **10**

Given :

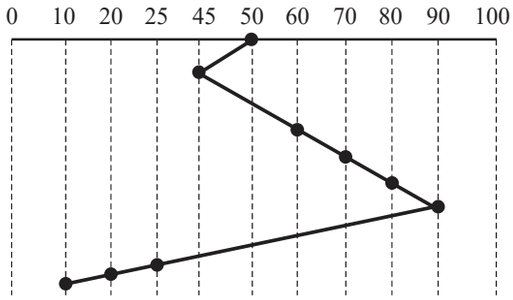
Number of tracks = 100

Disk access request :

45, 20, 90, 10, 50, 60, 80, 25, 70

Initial position of head = 50

SSTF :

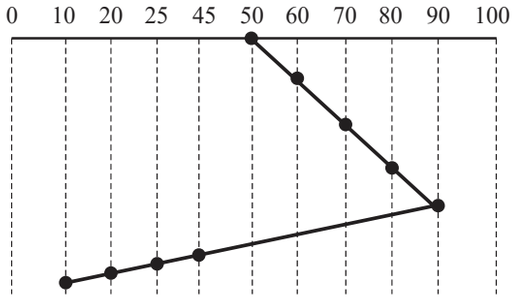


SSTF distance

$= (50-45)+(90-45)+(90-10)$

$= 5+45+80 = 130$

SCAN :



SCAN distance $= (90-50)+(90-10)$

$= 40+80 = 120$

$\therefore$ SSTF distance – SCAN distance

$= 130-120 = 10$

Hence, the correct answer is 10.

9 **(A)**

Given :

Allocation			
	E	F	G
P_0	1	0	1
P_1	1	1	2
P_2	1	0	3
P_3	2	0	0

Max			
	E	F	G
P_0	4	3	1
P_1	2	1	4
P_2	1	3	3
P_3	5	4	1

Method 1

Current available = $(E, F, G) = (3, 3, 0)$

After P_0, allocation of P_0 added to available

So, now available becomes $\Rightarrow \begin{matrix} E & F & G \\ 4 & 3 & 1 \end{matrix}$

With this available P_0 added to available. So now after P_2, available is $\Rightarrow \begin{matrix} E & F & G \\ 5 & 3 & 4 \end{matrix}$

With this available P_1 can complete. Hence after P_1, available is $\Rightarrow \begin{matrix} E & F & G \\ 6 & 4 & 6 \end{matrix}$

At the end P_3 also can complete.

Hence the system is in safe state.

$\therefore$ Hence A is correct option.

Method 2

	Max	Allocation	Need
	E F G	E F G	E F G
P_0	4 3 1	1 0 1	3 3 0
P_1	2 1 4	1 1 2	1 0 2
P_2	1 3 3	1 0 3	0 3 0
P_3	5 4 1	2 0 0	3 4 1

Available = $(E, F, G) = (3, 3, 0)$

Safe sequence:

(P_0, P_2, P_1, P_3)

$P_0 : P_0$ can be allotted (3, 3, 0)

After completion available = (4, 3, 1)

$P_2 : P_2$ can be allotted (0, 3, 0)

After completion available = (5, 3, 4)

$P_1 : P_1$ can be allotted (1, 0, 2)

After completion available = (6, 4, 6)

$P_3 : P_3$ can be allotted (3, 4, 1)

After completion available = (8, 4, 6)

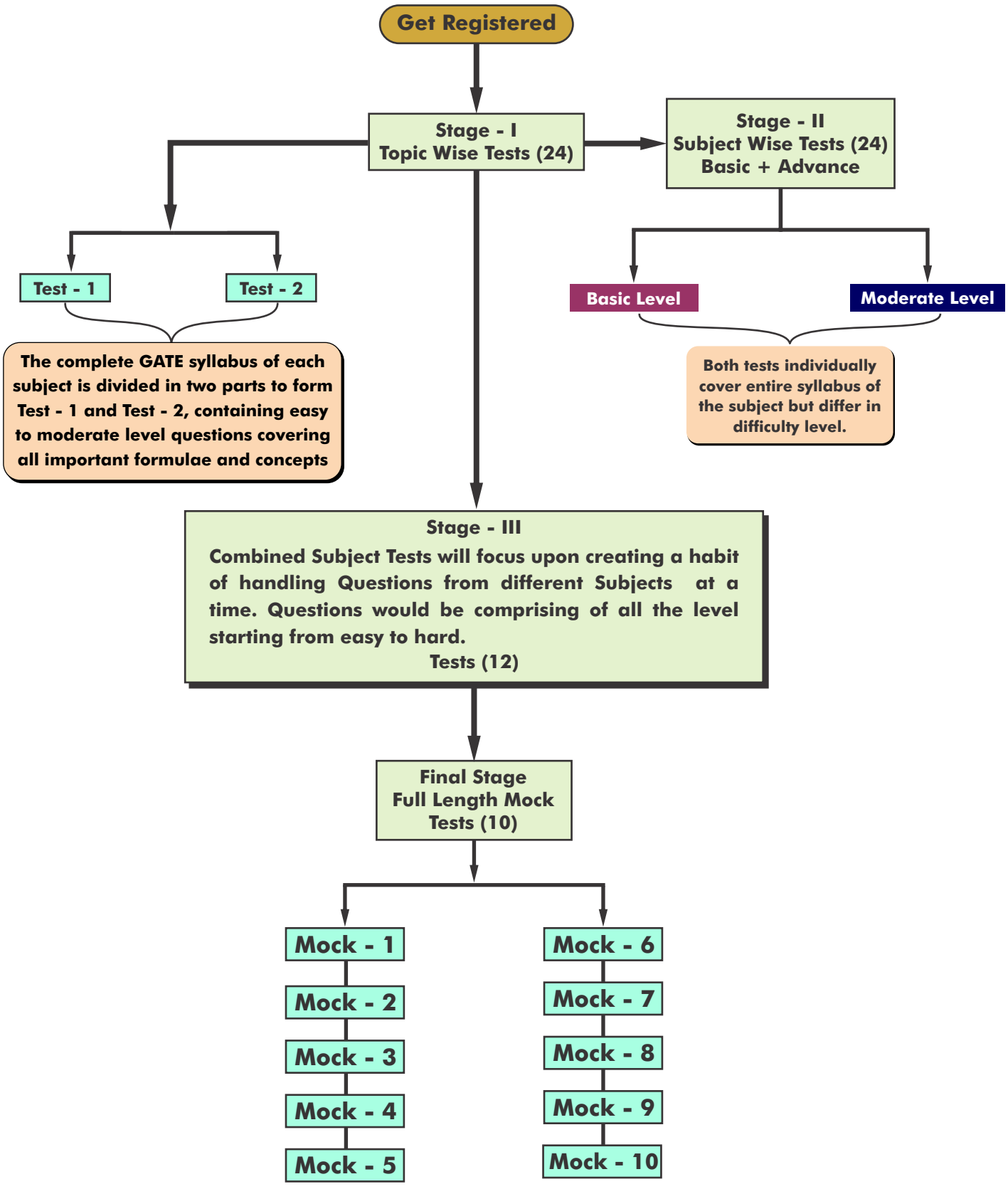
Hence, the correct option is (A).

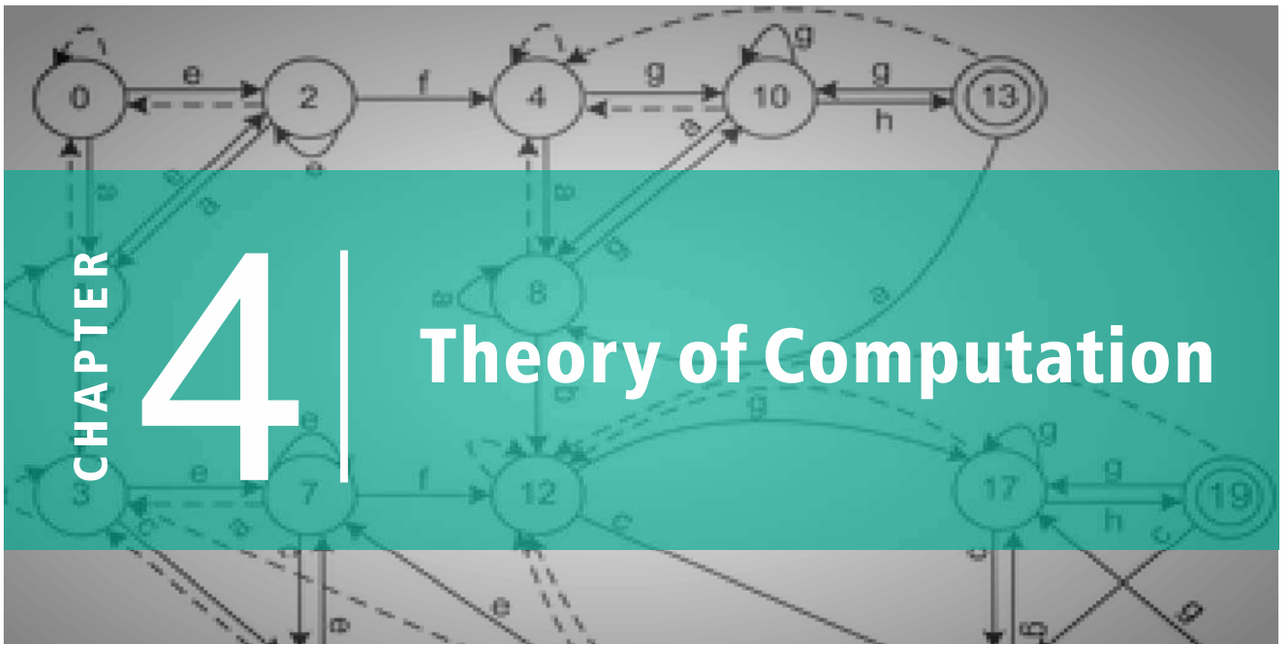
❖❖❖❖

GATE ACADEMY

TEST SERIES STRUCTURE (CS-IT)

(One Year Program)





CHAPTER 4 Theory of Computation

Marks Distribution of Theory of Computation in Previous Year GATE Papers.

Exam Year	1 Mark Ques.	2 Marks Ques.	Total Marks
2003	3	6	15
2004*	1	4	9
2005*	-	7	14
2006*	2	5	12
2007*	2	5	12
2008*	3	5	13
2009	4	3	10
2010	1	3	7
2011	3	3	9
2012	4	1	6
2013	2	3	8
2014 Set-1	2	2	6
2014 Set-2	2	2	6
2014 Set-3	2	2	6

Exam Year	1 Mark Ques.	2 Mark Ques.	Total Marks
2015 Set-1	1	2	5
2015 Set-2	1	2	5
2015 Set-3	1	1	3
2016 Set-1	3	3	9
2016 Set-2	3	4	11
2017 Set-1	2	4	10
2017 Set-2	3	3	9
2018	2	3	8
2019	2	3	8
2020	3	3	9
2021 Set-1	2	3	8
2021 Set-2	3	3	9

*** CS and IT combined**

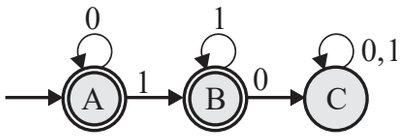
Syllabus : Theory of Computation

Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and contex-free languages, pumping lemma. Turing machines and undecidability.

Contents : Theory of Computation

1994 IIT Kharagpur

1 The regular expression for the language recognized by the finite state automaton of figure is _____.



2003 IIT Madras

2 Let $G = (\{S\}, \{a,b\}, R, S)$ be a context free grammar where the rule set R is

$$S \rightarrow aSb \mid S\,S \mid \varepsilon$$

Which of the following statements is true?

(A) G is not ambiguous

(B) There exist $x, y \in L(G)$ such that $xy \notin L(G)$

(C) There is a deterministic pushdown automaton that accepts $L(G)$

(D) We can find a deterministic finite state automaton that accepts $L(G)$

3 A single tape Turing Machine M has two states q_0 and q_1 of which q_0 is the starting state. The tape alphabet of M is $\{0,1,B\}$ and its input alphabet is $\{0,1\}$. The symbol B is the blank symbol used to indicate end of an input string. The transition function of M is described in the following table.

	0	1	B
q_0	$q_1,1,R$	$q_1 1,R$	Halt
q_1	$\langle M,w,i \rangle$	$q_1,1,R$	q_0,B,L

The table is interpreted as illustrated below.

The entry $(q_1,1,R)$ in row q_0 and column 1 signifies that if M is in state q_0 and reads 1 on the current tape square, then it writes 1 on the same tape square, moves its tape head one position to the right and transitions to state q_1. Which of the following statement is true about M?

(A) M does not halt on any string in $(0+1)^+$

(B) M does not halt on any string in $(00+1)^*$

(C) M halts on all string ending in a 0

(D) M halts on all string ending in a 1

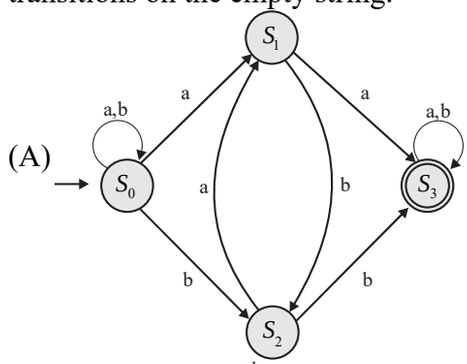
2007 IIT Kanpur

4 Consider the regular expression

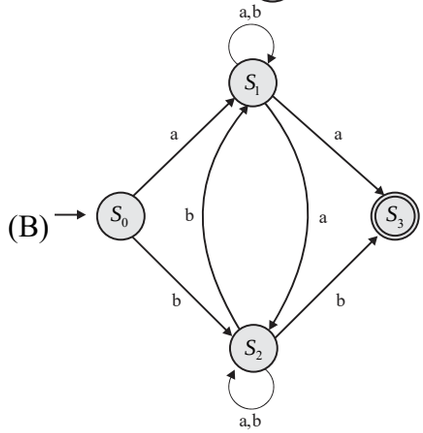
$$R = (a+b)^*(aa+bb)(a+b)^*$$

Which of the following non-deterministic finite automata recognizes the language defined by the regular expression R ? Edges labelled λ denote transitions on the empty string.

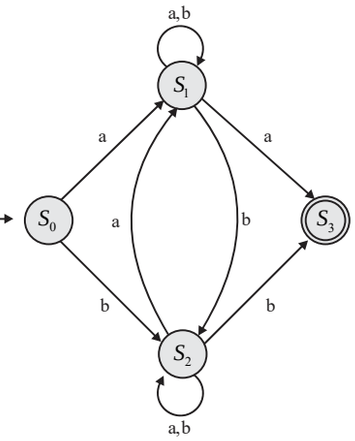
(A)



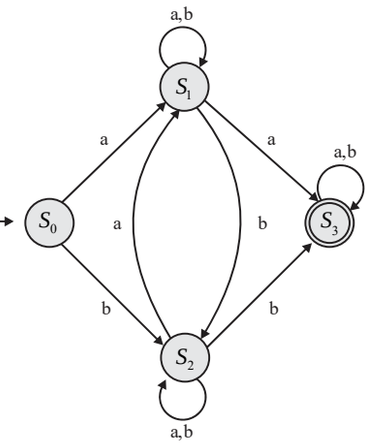
(B)



(C)



(D)



2009 IIT Roorkee

5 Which one of the following languages over the alphabet {0, 1} is described by the regular expression (0 +1)*0(0+1)* 0(0+1)*

(A) The set of all strings containing the substring 00

(B) The set of all strings containing at most two 0's

(C) The set of all strings containing at least two 0's

(D) The set of all strings that begin and end with either 0 or 1

2011 IIT Madras

6 Consider the languages L_1, L_2 and L_3 as given below.

$L_1 = \{0^p 1^q \mid p, q \in N\}$

$L_2 = \{0^p 1^q \mid p, q \in N \text{ and } p = q\}$ and

$L_3 = \{0^p 1^q 0^r \mid p, q, r \in N \text{ and } p = q = r\}$

Which of the following statements is not TRUE?

(A) Pushdown automata (PDA) can be used to recognize L_1 and L_2

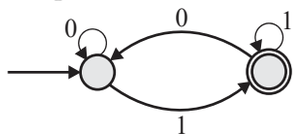
(B) L_1 is a regular language

(C) All the three languages are context free

(D) Turing machines can be used to recognize all the languages.

2014 IIT Kharagpur

7 Which of the regular expression given below represent the following DFA?



1. $0*1(1+00*1)*$
2. $0*1*1+11*0*1$
3. $(0+1)*1$

(A) 1 and 2 only (B) 1 and 3 only

(C) 2 and 3 only (D) 1, 2, and 3

2017 IIT Roorkee

8 If G is a grammar with productions

$$S \rightarrow SaS \mid aSb \mid bSa \mid SS \mid \varepsilon$$

Where S is the start variable, then which one of the following string is not generated by G?

(A) *abab* (B) *aaab*

(C) *abbaa* (D) *babba*

2018 IIT Guwahati

9 Consider the following languages :

I. $\left\{ a^m b^n c^p d^q \mid m + p = n + q, \text{ where } m, n, p, q \geq 0 \right\}$

II. $\left\{ a^m b^n c^p d^q \mid m = n \text{ and } p = q, \text{ where } m, n, p, q \geq 0 \right\}$

III. $\left\{ a^m b^n c^p d^q \mid m = n = p \text{ and } p \neq q, \text{ where } m, n, p, q \geq 0 \right\}$

IV. $\left\{ a^m b^n c^p d^q \mid mn = p + q, \text{ where } m, n, p, q \geq 0 \right\}$

Which of the languages above are context-free?

(A) I and IV only

(B) I and II only

(C) II and III only

(D) II and IV only

2021 IIT Bombay

10 Consider the following two statements about regular languages:

S1: Every infinite regular language contains an undecidable language as a subset.

S2: Every finite language is regular.

Which one of the following choices is correct?

(A) Only S1 is true

(B) Only S2 is true

(C) Both S1 and S2 are true

(D) Neither S1 nor S2 is true

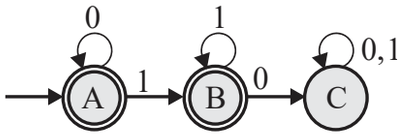
Answers Theory of Computation

1.	*	**2.**	C	**3.**	A	**4.**	A	**5.**	C
6.	C	**7.**	B	**8.**	D	**9.**	B	**10.**	C

Explanations **Theory of Computation**

1 (*)

Given :



Method 1

Using Arden's theorem :

Step 1 : Form an equation for each state

$$A = \varepsilon + A.0 \quad \ldots (i)$$

$$B = A.1 + B.1 \quad \ldots (ii)$$

$$C = B.0 + C.0 + C.1 \quad \ldots (iii)$$

Step 2 : Bring final state in the form $R = Q + RP$

Equation (i) is already in the form $R = Q + RP$

Comparing equation (i) with $R = Q + RP$ we get

$R = A,\ Q = \varepsilon,\ P = 0$

Since $R = Q + RP$ has a unique solution i.e. $R = QP^*$

Therefore A can be written as

$$A = \varepsilon \cdot 0^*$$

$$A = 0^*$$

Similarly for state B (since B is a final state)

Equation (ii) is $B = A.1 + B.1$

Comparing equation (ii) with $R = Q + RP$ we get

$$R = B,\ Q = A.1,\ P = 1$$

Therefore, B can be written as

$$B = (A \cdot 1) \cdot 1^*$$

$$B = (0^*1) \cdot 1^*$$

$$B = 0^*11^*$$

So final regular expression will be

$$A + B = 0^* + 0^*11^*$$

$$A + B = 0^*(\varepsilon + 11^*)$$

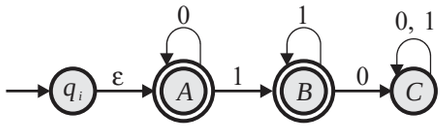
$$A + B = 0^*(\varepsilon + 1^+)$$

$$A + B = 0^*1^*$$

Method 2

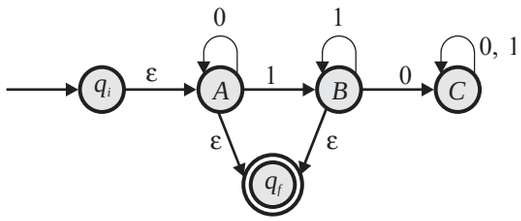
Using state elimination method :

Step 1 : If there exist any incoming edge to the initial state, then create a new initial state having no incoming edge to it. Therefore, the new FSA will be



Step 2 : If there exists multiple final states in the DFA, then convert all the final states into non-final states and create a new single final state.

Therefore, the new FSA will be

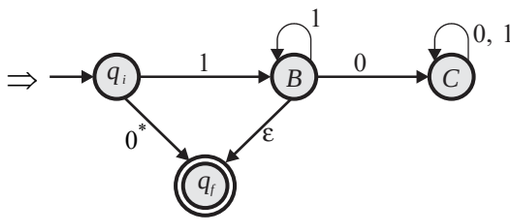


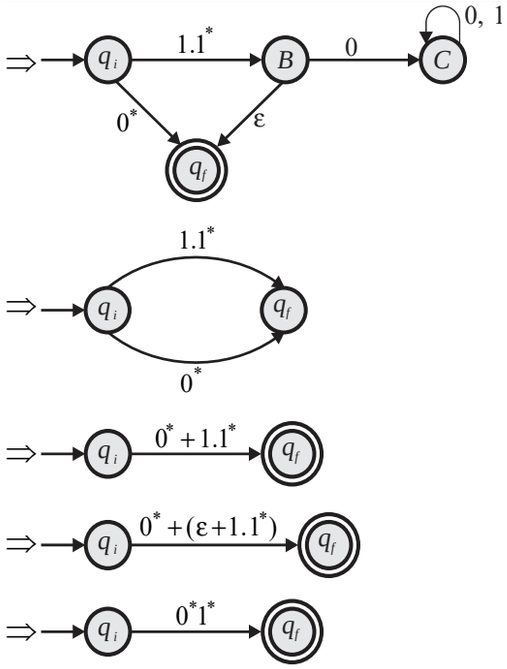
Step 3 : If there exist any outgoing edge, then create a new final state having no outgoing edge from it.

Above FSA already satisfies this condition.

Step 4 : Eliminate all the intermediate states.

Therefore the final FSA will be,





Hence, the regular expression will be 0^*1^*.

2 (C)

Given :

$$G = (\{S\}, \{a,b\}, R, S)$$

$$S \to aSb \mid SS \mid \varepsilon$$

Method 1

$S \to aSb \mid SS \mid \varepsilon$

Language generated by this grammar is

$$L(G) = \{\varepsilon, ab, abab, aabb, abaabb, ababab, aabbab, aaabbb,\}$$

$\Rightarrow$ $L(G) = \{w \mid w$ has equal number of a and b and w starts with a only$\}$

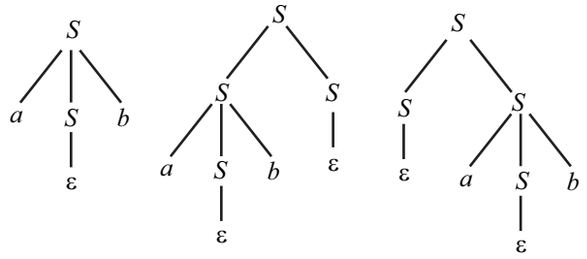
From options :

Option (A) : G is not ambiguous.

To check whether G is ambiguous or not we have to check if there exists multiple parse tree (derivation tree) of a string accepted by G.

Consider string ab,

ab can be derived as.



So, there exist multiple derivation tree for string ab therefore G is ambiguous, Hence option (A) is false.

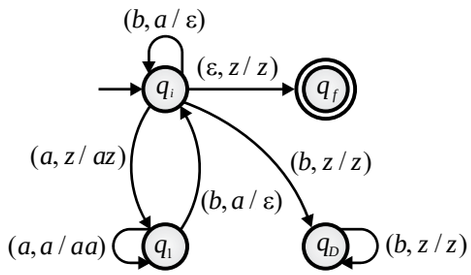
Option (B) : There exist $x, y \in L(G)$ such that $xy \notin L(G)$.

Since, $x \in L(G)$, which implies that x has equal number of a and b; similarly, $y \in L(G)$, which implies that y has equal number of a and b.

$\Rightarrow$ Concatenating x and y will result in xy and will contain equal number of a and b, hence $xy \in L(G)$.

Option (C) : There is a deterministic pushdown automata that accepts $L(G)$.

The DPDA for $L(G)$ can be constructed as follows:



Therefore, option (C) is true.

Option (D) : We can find a deterministic finite automata that accepts $L(G)$.

Since, the language $L(G)$ requires comparison, therefore, a memory element is required, So DFA cannot be constructed for the same.

Hence, the correct option is (C)

Method 2

It will become a lot easier to analyze the given problem if we replace terminal a and b by '(' and ')' respectively.

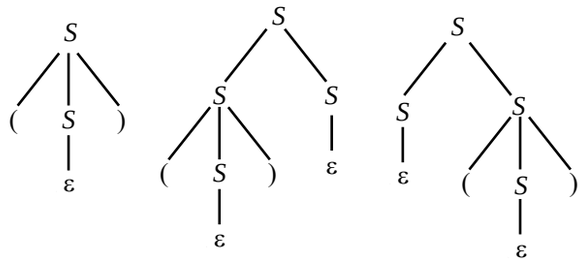
$S \to (S) \mid SS \mid \varepsilon$

$L(G)$ = balanced parentheses [each left parenthesis has a matching right parenthesis and are well nested]

Example : ()(),(()),(()()()).

Consider each option one by one :

Option (A) :



String () can be derived by above three way each having different derivation tree. So G is Ambiguous.

So, (A) is false.

Option (B) :

Concatenation of two balance parenthesis will be balanced also, equation x = (()), y = (), xy = (())().

So, (B) is false.

Option (C) :

We can design Deterministic PDA for L, push left parenthesis only in stack and pop with right parenthesis.

So, (C) is true.

Option (D) :

We cannot design DFA for L because we need a stack to match left parenthesis with right parenthesis.

So, (D) is false.

Hence, the correct option is (C).

3 (A)

Given :

A single tape Turing Machine

Set of states $= \{q_0, q_1\}$

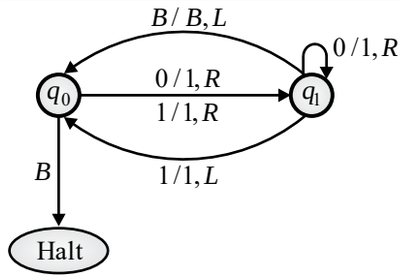
Initial state $= q_0$

Tape alphabet $= \{0, 1, B\}$

Input $= \{0, 1\}$

Transition function

	0	1	B
q_0	$q_1, 1, R$	$q_1, 1, R$	Halt
q_1	$q_1, 1, R$	$q_0, 1, L$	q_0, B, L



The given Turing machine will halt only for ε.

Hence, the correct option is (A).

4 (A)

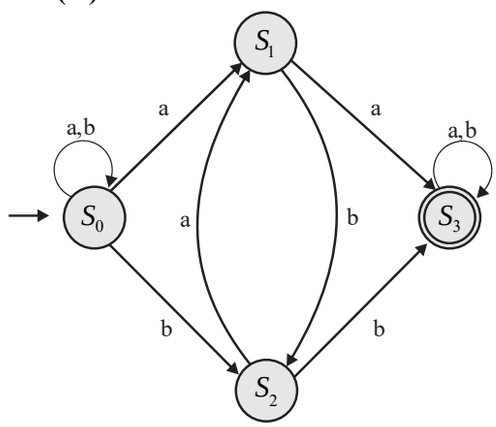
Given : Regular expression,

$$R = (a+b)^*(aa+bb)(a+b)^*$$

$\Rightarrow$ $L(R) =$ string should contain either aa or bb

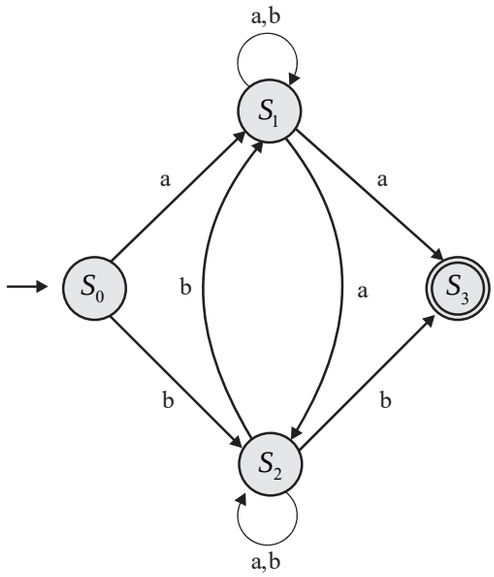
Consider each option one by one :

Option (A) :



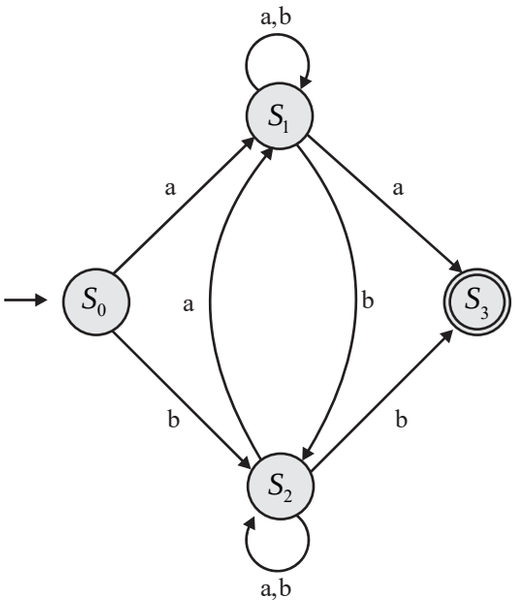
It accepts any string containing either *aa* or *bb* and it does not accept any other string.

Option (B) :



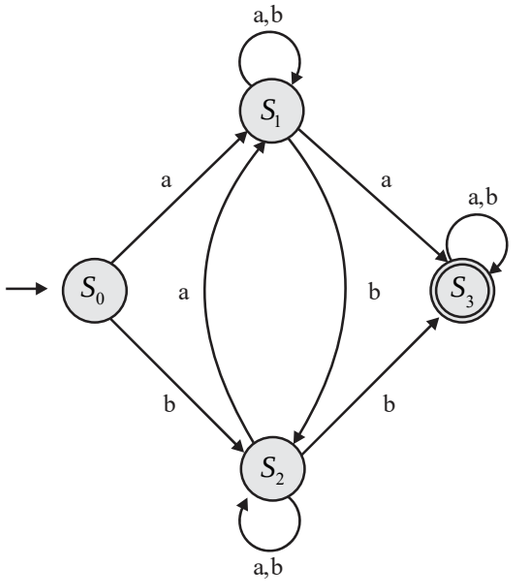
It accepts '*aba*', so B is incorrect.

Option (C) :



It accepts '*aba*', so C is incorrect.

Option (D) :



It accepts '*a*', so D is incorrect.

Hence, the correct option is (A).

5 (C)

Given : regular expression

$$(0+1)^*0(0+1)^*0(0+1)^*$$

Language generated by this regular expression is

$$L=\{00,000,100,010,001,0000,....\}$$

$L=\{w \mid w\in\{0,1\}^*, w \text{ contains at least two 0's}\}$

Hence, the correct option is (C).

6 (C)

Given :

$$L_1=\{0^p1^q \mid p,q\in N\}$$

$$L_2=\{0^p1^q \mid p,q\in N \text{ and } p=q\} \text{ and}$$

$$L_3=\{0^p1^q0^r \mid p,q,r\in N \text{ and } p=q=r\}$$

Conclusions –

1. Since, p and q are independent therefore, L_1 is Regular.
2. Since, only one comparison is performed between p and q therefore, L_2 is CFL.

3. Since, more than one comparison is performed between p and q therefore, L_3 is CSL.

Clearly, option (C) is incorrect. Since all the three languages are not CFL.

Consider each option one by one :

Option (A) : Pushdown automata (PDA) can be used to recognize L_1 and L_2

L_1 is Regular therefore, CFL too. L_2 is CFL. So, PDA can be designed to recognize L_1 and L_2.

Therefore (A) is true.

Option (B) : L_1 is a regular language

From conclusion 1, L_1 is a regular language.

Therefore (B) is true.

Option (C) : All the three languages are context free

L_1 is Regular therefore, CFL too, L_2 is CFL, but L_3 is CSL.

Therefore (C) is false.

Option (D) : Turing machines can be used to recognize all the languages.

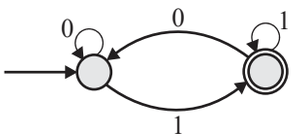
Turing machine recognizes all type-0 language. So, it can recognize L_1, L_2 and L_3.

Therefore (D) is true.

Hence, the correct option is (C).

7 (B)

Given : DFA



Given regular expressions are ;

1. $0*1(1+00*1)*$
2. $0*1*1+11*0*1$
3. $(0+1)*1$

Method 1

Using Arden's theorem

$$q_0 = \varepsilon + q_0 \cdot 0 + q_1 \cdot 0$$

$$q_0 = (\varepsilon + q_1 \cdot 0) \cdot 0*$$

$$q_0 = q_0 \cdot 1 + q_1 \cdot 1$$

$$R.E. = (\varepsilon + q_1 0) \cdot 0*1 + q_1 \cdot 1$$

$$R.E. = \varepsilon \cdot 0*1 + q_1 \cdot 00*1 + q_1 \cdot 1$$

$$= 0*1 + q_1 \cdot (00*1 + 1)$$

$$= 0*1 + (00*1 + 1)*$$

Reducing above regular expression

$$= 0*1((00* + \varepsilon)1)*$$

It produces the strings like

$$1, 01, 11, 001, 111, 101, \ldots$$

$\therefore$ Strings that are ending with 1

$$= (0+1)*1 \cong 0*1(00*1+1)*$$

$\therefore$ I and III are correct.

Checking II : $0*1*1+11*0*1$

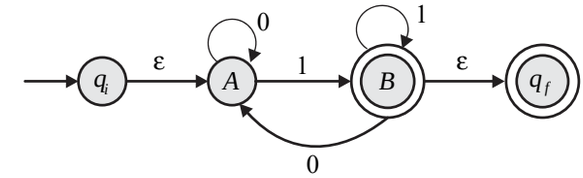
We can't generate strings like $0101, 11011, \ldots$

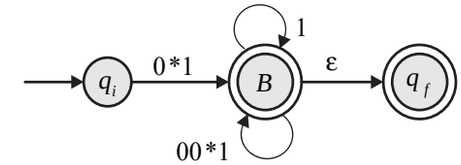
$\therefore$ Only I and III are correct option.

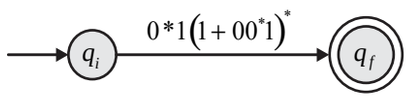
Hence, the correct option is (B).

Method 2

State Elimination method







Hence, the correct option is (B).

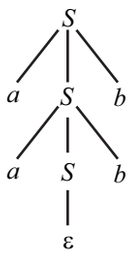
8 (D)

To check if a string is generated by the grammar, we need to check if a parse tree can be constructed or not.

Consider each option one by one :

Option (A) : *abab*

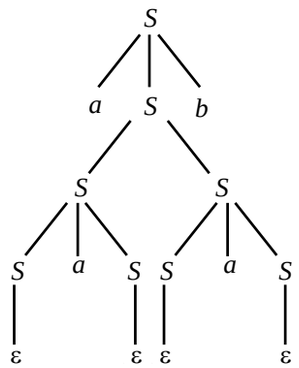
Parse tree to generate *abab* -



Therefore, we can conclude that *abab* can be generated by G.

Option (B) : *aaab*

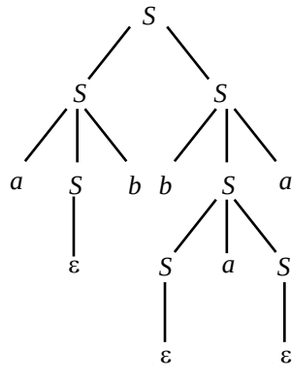
Parse tree to generate *aaab* -



Therefore, we can conclude that *aaab* can be generated by G.

Option (C) : *abbaa*

Parse tree to generate *abbaa* -



Therefore, we can conclude that *abbaa* can be generated by G.

Option (D) : *babba*

Since, we cannot construct parse tree for this string,

Hence, we can conclude that *babba* cannot be generated by G.

Hence the correct option is (D).

9 (B)

Given :

I. $\left\{a^m b^n c^p d^q \mid m+p=n+q, \text{ where } m,n,p,q \geq 0\right\}$

II. $\left\{a^m b^n c^p d^q \mid m=n \text{ and } p=q, \text{ where } m,n,p,q \geq 0\right\}$

III. $\left\{a^m b^n c^p d^q \mid m=n=p \text{ and } p \neq q, \text{ where } m,n,p,q \geq 0\right\}$

IV. $\left\{a^m b^n c^p d^q \mid mn=p+q, \text{ where } m,n,p,q \geq 0\right\}$

Consider $\left\{a^m b^n c^p d^q \mid m+p=n+q, \text{ where } m,n,p,q \geq 0\right\}$

Grammar for above language is –

$S \rightarrow aSd \mid ABC \mid \varepsilon$

$A \rightarrow aAb \mid ab \mid \varepsilon$

$B \rightarrow bBc \mid bc \mid \varepsilon$

$C \rightarrow cCd \mid cd \mid \varepsilon$

Above Grammar is CFG so it will generate CFL.

Consider $\left\{a^m b^n c^p d^q \mid m=n \text{ and } p=q, \text{ where } m,n,p,q \geq 0\right\}$

Grammar for above language is –

$S \rightarrow AB \mid \varepsilon$

$A \rightarrow aAb \mid ab$

$B \rightarrow cBd \mid cd$

Above Grammar is CFG so it will generate CFL.

Consider $\left\{ a^m b^n c^p d^q \mid m = n = p \text{ and } p \neq q, \text{ where } m, n, p, q \geq 0 \right\}$

Here more than one comparison is performed so it is not CFL.

Consider $\left\{ a^m b^n c^p d^q \mid mn = p + q, \text{ where } m, n, p, q \geq 0 \right\}$

Here more than one comparison is performed so it is CSL but NOT CFL.

Hence, the correct option is (B).

10 (C)

Consider each statement

S1: Every infinite regular language contains an undecidable language as a subset.

True. We can construct a subset N of A that we will prove non-regular by using pumping lemma.

S2: Every finite language is regular.

True. Every finite language is Regular. Because we can draw DFA for it.

Hence the correct option is (C).

❖❖❖❖

Finally... How to get registered for Test Series ?

Step - I	Step - II
Visit http://onlinetestseries.gateacademy.co.in Click on New Registration Fill the Form, You will be Redirected to the Test Portal Click on buy Package Make Payment using Coupon code (For gate academy student) or by online payment.	Send completely filled registration form to our office (for gate academy student)

Join GATE ACADEMY

Live Classroom/VOD/Pen Drive/G-Drive/ Tablet Course

For Assured Results

Full Course

GATE 2021-22

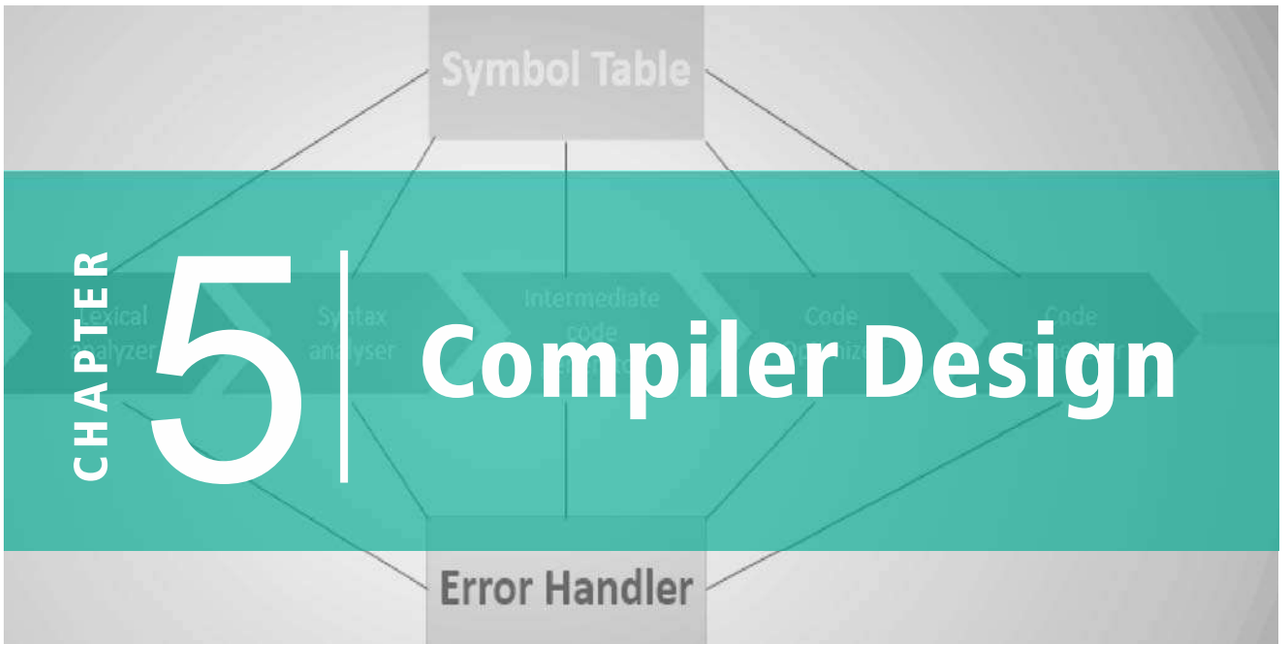
Gate Academy Live app

GET IT ON Google Play

GATE ACADEMY®

steps to success....

HEAD OFFICE : A / 114-115, SMRITI NAGAR, BHILAI - 490 020 (C.G.)
Contact : 97131-13156, 62662-02387, 0788-4034176
E-mail : info@gateacademy.co.in



CHAPTER 5

Compiler Design

Marks Distribution of Compiler Design in Previous Year GATE Papers.

Exam Year	1 Mark Ques.	2 Marks Ques.	Total Marks
2003	3	5	13
2004*	3	1	5
2005*	1	4	9
2006*	1	5	11
2007*	1	5	11
2008*	2	1	4
2009	1	-	1
2010	2	1	4
2011	1	1	3
2012	-	2	4
2013	1	3	7
2014 Set-1	1	1	3
2014 Set-2	2	2	6
2014 Set-3	2	1	4

Exam Year	1 Mark Ques.	2 Mark Ques.	Total Marks
2015 Set-1	1	2	5
2015 Set-2	2	1	4
2015 Set-3	1	1	3
2016 Set-1	1	2	5
2016 Set-2	1	1	3
2017 Set-1	2	1	4
2017 Set-2	2	1	4
2018	1	2	5
2019	2	2	6
2020	2	1	4
2021 Set-1	2	3	8
2021 Set-2	2	2	6

*** CS and IT combined**

Syllabus : Compiler Design

Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimisation, Data flow analyses: constant propagation, liveness analysis, common subexpression elimination.

Contents : Compiler Design

1996 IISc Bangalore

1 The pass number for each of the following activities

(i) Object code generation

(ii) Literals added to literal table

(iii) Listing printed

(iv) Address resolution of local symbols that occur in a two assembler respectively are

(A) 1, 2, 1, 2 (B) 2, 1, 2, 1

(C) 2, 1, 1, 2 (D) 1, 2, 2, 2

2008 IISc Bangalore

2 Which of the following statements are true?

I. Every left-recursive grammar can be converted to a right-recursive grammar and vice-versa

II. All ε productions can be removed from any context-free grammar by suitable transformations

III. The language generated by a context-free grammar all of whose productions are of the form $X \rightarrow w$ or $X \rightarrow wY$ (where, w is a string of terminals and Y is a non-terminal), is always regular

IV. The derivation trees of strings generated by a context-free grammar in Chomsky Normal Form are always binary trees

(A) I, II, III and IV

(B) II, III and IV only

(C) I, III and IV only

(D) I, II and IV only

2009 IIT Roorkee

3 Which of the following statements are TRUE?

I. There exist parsing algorithms for some programming languages whose complexities are less than $\Theta(n^3)$.

II. A programming language which allows recursion can be implemented with static storage allocation.

III. No L-attributed definition can be evaluated in the framework of bottom-up parsing.

IV. Code improving transformations can be performed at both source language and intermediate code level.

(A) I and II

(B) I and IV

(C) III and IV

(D) I, III and IV

2014 IIT Kharagpur

4 Consider the grammar defined by the following production rules, with two operators* and +

$S \rightarrow T * P$

$T \rightarrow U \mid T * U$

$P \rightarrow Q + P \mid Q$

$Q \rightarrow id$

$U \rightarrow id$

Which one of the following is TRUE?

(A) + is left associative, while * is right associative

(B) + is right associative, while * is left associative

(C) Both + and * are right associative

(D) Both + and * are left associative

5 Consider the basic block given below.

$a = b + c$

$c = a + d$

$d = b + c$

$e = d - b$

$a = e + b$

The minimum number of nodes and edges present in the DAG representation of the above basic block respectively are

(A) 6 and 6 (B) 8 and 10

(C) 9 and 12 (D) 4 and 4

2018 IIT Guwahati

6 Which one of the following statements is FALSE?

(A) Context-free grammar can be used to specify both lexical and syntax rules.

(B) Type checking is done before parsing.

(C) High-level language programs can be translated to different Intermediate Representations.

(D) Arguments to a function can be passed using the program stack.

Answers Compiler Design

1.	B	2.	C	3.	B	4.	B	5.	A
6.	B								

Explanations Compiler Design

1 (B)

If compilation is done at a single time completely, then it is called as single pass compiler otherwise it is called as multipass compiler.

Key Point

In Multipass compiler, source code is processed several times.

Example in two pass compiler:

(1) In first pass symbol table is created, literals are added to literal table and address resolution of local symbols is done.

(2) In second pass, it translates the program into machine language. So, object code generation and listing printed is done at second pass.

Key Point

(1) Single pass complier
Advantages : Less time
Disadvantages : More space

(2) Multipass compiler
Advantages : Less space
Disadvantages : More time

Hence, the correct option is (B).

2 (C)

Given:

Statement I : Every left-recursive grammar can be converted to a right-recursive grammar and vice-versa.

True, using GNF we can convert Left recursive grammar to right recursive and by using reversal of CFG and GNF we can convert right recursive to left recursive.

Statement II : All productions can be removed from any context-free grammar by suitable transformations.

False, because ε cannot be removed always because if ε is in the language then we can't remove ε production from Start symbol.

Statement III : The language generated by a context-free grammar all of whose productions are of the form $X \rightarrow w$ or $X \rightarrow wY$ (where, w is a string of terminals and Y is a non-terminal), is always regular.

True, since, every right linear grammar always generates regular language.

Statement IV : The derivation trees of strings generated by a context-free grammar in Chomsky Normal Form are always binary trees.

True, only two non-terminals are there in each production in CNF. So, it will always form a binary tree.

Hence, the correct option is (C).

3 (B)

Given :

I. There exist parsing algorithms for some programming languages whose complexities are less than $\Theta(n^3)$.

Yes there does exist parsing algorithms which run in $\Theta(n^3)$ so, I is true.

II. A programming language which allows recursion can be implemented with static storage allocation.

In recursion, compiler cannot determine the space needed for recursive calls so recursion cannot be implemented with static storage allocation. It needs dynamic memory allocation. So, II is false.

III. No L-attributed definition can be evaluated in the framework of bottom-up parsing.

Every S-attributed definition is also an L-attributed definition and can be evaluated in the framework of bottom up parsing. So, III is false.

IV. Code improving transformations can be performed at both source language and intermediate code level.

IV is true. Code improving transformations can be performed at both source language and intermediate code level.

Hence, the correct option is (B).

4 (B)

Given :

$S \rightarrow T * P$

$T \rightarrow U \mid T * U$

$P \rightarrow Q + P \mid Q$

$Q \rightarrow id$

$U \rightarrow id$

Method 1

$T \rightarrow T * U$

T is generating $T * U$ recursively, so * is left associative.

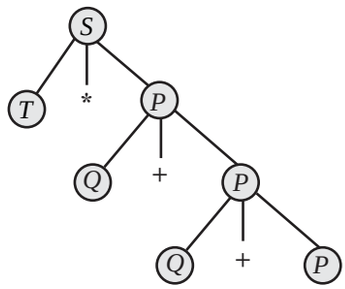
Similarly,

$P \rightarrow Q + P$

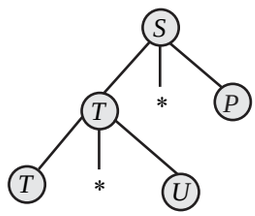
Right recursion is there, so + is right associative.

Method 2

Draw parse tree and check.



+ is right associative.



* is left associative

Hence, the correct option is (B).

5 (A)

Given :

Basic block

$$a = b + c$$
$$c = a + d$$
$$d = b + c$$
$$e = d - b$$
$$a = e + b$$

$a = b + c \Rightarrow a = e + b$

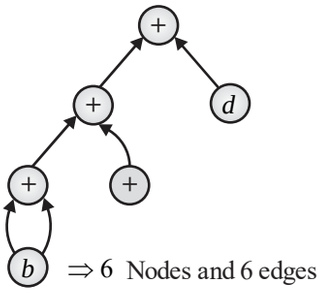
$c = a + d \Rightarrow a = d - b - b = d$

$d = b + c \Rightarrow a = b + c$

$e = d - b \Rightarrow a = b + a + d$

$a = e + b \Rightarrow a = b + b + c + d$

$\therefore$ b + b + c + d is final expression



$\Rightarrow$ 6 Nodes and 6 edges

Hence, the correct option is (A).

6 (B)

Consider each options one by one :

Option (A) : Context free grammar can be used to specify both lexical and syntax rules.

Since lexical rules are nothing but regular expressions we can use CFGs to represent such rules (Every Type-3 grammar is Type-2 grammar). Additionally, syntax rules can be represented by CFGs.

Option (B) : Type checking is done before parsing.

Type checking is done during semantic analysis phase which comes after parsing.

Option (C) : High level language programs can be translated to different intermediate representations.

We have various types of intermediate code representations, ex. 3-address code, postfix notations, syntax trees.

Option (D) : Arguments to a function can be passed using the program stack.

Program stack holds the activation record of the function called, which stores function parameters, return value, return address etc.

Hence, the correct option is (B).

❖❖❖❖

12 YEARS GATE SOLVED PAPERS

4 Years ESE Solved Papers

GATE 2022

GENERAL APTITUDE

UMESH DHANDE

AE AG AR BM BT CY CE CH CS EC EE IN ME PI ES EY GG MA MN MT PE PH ST TF XE XH XL

Topic-wise Previous 12 Years GATE Solutions

- Diligent solutions of GATE previous year questions (2010-2021)
- Multi method approach for a single problem to develop crystal clear concepts
- Video solution for conspicuous questions to enhance problem solving skills

GATE ACADEMY®
steps to success...

CHAPTER 6 Computer Network

Marks Distribution of Computer Network in Previous Year GATE Papers.

Exam Year	1 Mark Ques.	2 Marks Ques.	Total Marks
2003	2	3	8
2004*	3	4	11
2005*	5	2	9
2006*	1	5	11
2007*	2	6	14
2008*	1	4	9
2009	-	5	10
2010	2	3	8
2011	2	2	6
2012	3	3	9
2013	3	2	7
2014 Set-1	2	3	8
2014 Set-2	3	2	7
2014 Set-3	3	3	9

Exam Year	1 Mark Ques.	2 Mark Ques.	Total Marks
2015 Set-1	4	2	8
2015 Set-2	2	3	8
2015 Set-3	2	3	8
2016 Set-1	2	4	10
2016 Set-2	3	4	11
2017 Set-1	2	3	8
2017 Set-2	3	2	7
2018	3	2	7
2019	2	4	10
2020	2	2	6
2021 Set-1	2	3	8
2021 Set-2	2	3	8

*** CS and IT combined**

Syllabus : Computer Network

Concept of layering: OSI and TCP/IP Protocol Stacks; Basics of packet, circuit and virtual circuit-switching; Data link layer: framing, error detection, Medium Access Control, Ethernet bridging; Routing protocols: shortest path, flooding, distance vector and link state routing; Fragmentation and IP addressing, IPv4, CIDR notation, Basics of IP support protocols (ARP, DHCP, ICMP), Network Address Translation (NAT); Transport layer: flow control and congestion control, UDP, TCP, sockets; Application layer protocols: DNS, SMTP, HTTP, FTP, Email.

Contents : Computer Network

2004 IIT Delhi

1 Which of the following is NOT true with respect to a transparent bridge and a router?

(A) Both bridge and router selectively forward data packets

(B) A bridge uses IP addresses while a router uses MAC addresses

(C) A bridge builds up its routing table by inspecting incoming packets

(D) A router can connect between a LAN and a WAN

2 Consider a parity check code with three data bits and four parity check bits.

Three of the Code Words are 0101011, 1001101 and 1110001.

Which of the following are also code words?

I. 0010111 II. 0110110

III. 1011010 IV. 0111010

(A) I and III

(B) I, II and III

(C) II and IV

(D) I, II, III and IV

2012 IIT Delhi

3 An Internet Service Provider (ISP) has the following chunk of CIDR-based IP addresses available with it: 245.248.128.0/20. The ISP wants to give half of this chunk of addresses to Organization *A*, and a quarter to Organization *B*, while retaining the remaining with itself. Which of the following is a valid allocation of addresses to *A* and *B*?

(A) 245.248.136.0/21 and 245.248.128.0/22

(B) 245.248.128.0/21 and 245.248.128.0/22

(C) 245.248.132.0/22 and 245.248.132.0/21

(D) 245.248.136.0/24 and 245.248.132.0/21

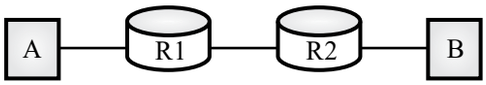
4 Consider an instance of TCP's Additive Increase Multiplicative Decrease (AIMD) algorithm where the window size at the start of the slow start phase is 2 MSS and the threshold at the start of the first transmission is 8 MSS. Assume that a timeout occurs during the fifth transmission. Find the congestion window size at the end of the tenth transmission.

(A) 8 MSS (B) 14 MSS

(C) 7 MSS (D) 12 MSS

2014 IIT Kharagpur

5 Consider the store and forward packet switched network given below. Assume that the bandwidth of each link is 10^6 bytes/sec. A user on host *A* sends a file of size 10^3 bytes to host *B* through routers R_1 and R_2 in three different ways. In the first case a single packet containing the complete file is transmitted from *A* to *B*. In the second case, the file is split into 10 equal parts, and these packets are transmitted from *A* to *B*. In the third case, the file is split into 20 equal parts and these packets are sent from *A* to *B*. Each packet contains 100 bytes of header information along with the user data. Consider only transmission time and ignore processing, queuing and propagation delays. Also assume that there are no errors during transmission. Let T_1, T_2 and T_3 be the times taken to transmit the file in the first, second and third case respectively. Which one of the following is CORRECT?



(A) $T_1 < T_2 < T_3$

(B) $T_1 > T_2 > T_3$

(C) $T_2 = T_3, T_3 < T_1$

(D) $T_1 = T_3, T_3 > T_2$

2017 IIT Roorkee

6 The values of parameters for the Stop-and-Wait ARQ protocol are as given below :

- Bit rate of the transmission channel = 1 Mbps.
- Propagation delay from sender to receiver = 0.75 ms.
- Time to process a frame = 0.25 ms.
- Number of bytes in the information frame = 1980.
- Number of bytes in the acknowledge frame = 20.
- Number of overhead bytes in the information frame = 20.

Assume there are no transmission errors. Then, the transmission efficiency (expressed in percentage) of the Stop-and-Wait ARQ protocol for the above parameters is _____ (correct to 2 decimal places).

2019 IIT Madras

7 Consider that 15 machines need to be connected in a LAN using 8-port Ethernet switches. Assume that these switches do not have any separate uplink ports. The minimum number of switches needed is _____.

2021 IIT Bombay

8 A TCP server application is programmed to listen on port number P on host S. A TCP client is connected to the TCP server over the network.

Consider that while the TCP connection was active, the server machine S crashed and rebooted. Assume that the client does not use the TCP keepalive timer.

Which of the following behaviors is/are possible?

(A) If the client was waiting to receive a packet, it may wait indefinitely.

(B) The TCP server application on S can listen on P after reboot.

(C) If the client sends a packet after the server reboot, it will receive a RST segment.

(D) If the client sends a packet after the server reboot, it will receive a FIN segment.

Answers Computer Network

1.	B	2.	A	3.	A	4.	C	5.	D
6.	89.34	7.	3	8.	A, B, C				

Explanations **Computer Network**

1 (B)

Consider each option one by one :

Option (A) : Both bridge and router selectively forward data packets

It is true. Both bridge and router selectively forward data packets. Both forward a data packet to other side it is really belongs to that side otherwise they drops that packet.

Option (B) : A bridge uses IP addresses while a router uses MAC addresses

It is false, because Router operate at Network Layer by IP address and bridge operate at data link layer by *MAC* address.

Option (C) : A bridge builds up its routing table by inspecting incoming packets

It is true. A bridge builds up its routing table by inspecting incoming packets.

Example : self-learning bridges.

Option (D) : A router can connect between a LAN and a WAN

It is true. A router can connect between a LAN and a WAN.

Example : router connecting home LAN to internet.

Key Point

A bridge operate at layer 2 (data link layer) hence it uses MAC address.

A router operate at network layer hence it uses ip address.

Hence, the correct option is (B).

2 (A)

Method 1

The given codewords are :

A = 0101011

B = 1001101

C = 1110001

A⊕B = 1100110 (Hamming distance is 4)

B⊕C = 0111100 (Hamming distance is 4)

A⊕C = 1011010(Hamming distance is 4)

So, in our given code system, all codewords are at a hamming distance of 4.

Now considering each option one by one:

(I) 0010111 = x (say)

A⊕ x = 111100 (Hamming distance =4)

B⊕x = 1011010 (Hamming distance =4)

C⊕x = 1100110 (Hamming distance =4)

This new codeword x has same hamming distance with the present codewords. Hence, this is a valid codeword.

(II) 0110110 = y (say)

A⊕y = 0011101 (Hamming distance = 4)

B⊕y = 1111011 (Hamming distance = 6)

codeword y has a hamming distance of 6 with codeword B.

So, Choice (II) is surely invalid.

(III) 1011010 = z (say)

A⊕F = 1110001 (Hamming distance = 4)

B⊕F = 0010111 (Hamming distance = 4)

C⊕F = 0101011 (Hamming distance = 4)

Yes, this is a valid codeword.

(IV) 0111010 = m (say)

A⊕m = 010011 (Hamming distance = 3)

This is an invalid codeword.

Method 2

To find the hamming distance between two strings, count the number of positions in which the two strings differ.

Hamming distance between A and B is

0	1	0	1	0	1	1
1	0	0	1	1	0	1

Hamming distance = 4

Hamming distance between B and C is

1	0	0	1	1	0	1
1	1	1	0	0	0	1

Hamming distance = 4

Hamming distance between A and C is

0	1	0	1	0	1	1
1	1	1	0	0	0	1

Hamming distance = 4

Key Point

- In all valid codewords, the hamming distance between any two valid codewords remains constant.
- For finding the hamming distance between any two codewords, perform XOR operation of both and find the number of 1's present in the result.

Hence, the correct option is (A).

3 (A)

Given :

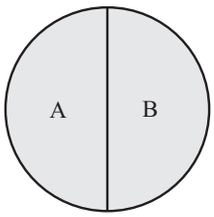
Chunk of CIDR based IP address-

245.248.128.0/20

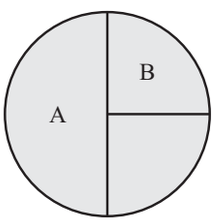
According to the question, this chunk is divided into three parts out of which half of the chunk is assigned to organization A, quarter of the chunk is divided to organization B.

Remaining is retained.

To divided a network into 3 subnets we need to first divide it into two halves.



For organization B, one of the half is divided into two halves.



So at first we borrow one bit from HID of given network, it will divide the network into two subnets

NID contains 20 bits,

HID contains 12 bits

IP address : 245.248.128.0/20

In binary : $245.248.\underbrace{10000000.0}_{HID} / 20$

For organization A, New IP address,

$245.248.10001000.0 \rightarrow 245.248.136.0 / 21$ … (i)

Or

$245.248.10000000.0 \rightarrow 245.248.128.0 / 21$ … (ii)

For organization B one more bit is borrowed from HID.

New IP address, if organization (A) uses (i)

$245.248.10000000.0 \rightarrow 245.248.128.00 / 22$

Or

$245.248.10000100.0 \rightarrow 245.248.132.0 / 22$

New IP address, if organization (A) uses (ii)

$245.248.10001000.0 \rightarrow 245.248.136.0 / 22$

Or

$245.248.10001100.0 \rightarrow 245.248.140.0 / 22$

Note : If putting 0 at 21st bit in A, then we have to put 1 at 21st bit in B.

If putting 1 at 21st bit in A, then we have to put 0 at 21st bit in B, to avoid conflict.

Hence, the correct option is (A).

4 (C)

Given :

Initial window size = 2 MSS

Threshold = 8 MSS

In slow start phase, window size increases exponentially.

$t = 1, \Rightarrow$ 2 MSS

$t = 2, \Rightarrow$ 4 MSS

$t = 3, \Rightarrow$ 8 MSS

After threshold additive increase, i.e. window size increases linearly.

$t = 4, \Rightarrow$ 9 MSS

$t = 5, \Rightarrow$ 10 MSS (fails)

During 5th transmission timeout occurs

New threshold, $Th_{new} = \frac{10}{2} = 5$

Again slow start phase :

$t = 6, \Rightarrow$ 1 MSS

$t = 7, \Rightarrow$ 2 MSS

$t = 8, \Rightarrow$ 4 MSS

$t = 9, \Rightarrow$ 5 MSS

$t = 10, \Rightarrow$ 6 MSS

So, at the end of 10^{th} successful transmission.

The congestion window size will be $(6+1) = 7$ MSS.

Hence, the correct option is (C).

5 (D)

Key Point

Transmission time = Packet size /Bandwidth

Given :

File size $= 1000$ bytes

Header size $= 100$ bytes

Bandwidth $= 10^6$ bytes/sec

Case I : Calculation of T_1 :

Transmission time for one link

$$= \frac{(1000+100)}{10^6}$$

$= 1100$ microsecond

Total time $= 3 \times 1100$

$= 3300$ microsecond

Case II : Calculation of T_2 :

Transmission time for one link for I[st] packet

$$= \frac{(100+100)}{10^6}$$

$= 200$ microsecond

Total time $= 3 \times 200 + 9 \times 200$

$= 2400$ micro second

Case III : Calculation of T_3 :

Transmission time for one link and one packet

$$= \frac{(50+100)}{10^6}$$

$= 150$ microsecond

Total time $= 3 \times 150 + 19 \times 150$

$= 3300$ microsecond

Hence, the correct option is (D).

6 89.34

Given :

Bit rate of the transmission channel

= 1 Mbps.

- Propagation delay from sender to receiver = 0.75 ms.
- Time to process a frame = 0.25 ms.
- Number of bytes in the information frame = 1980.
- Number of bytes in the acknowledge frame = 20.
- Number of overhead bytes in the information frame = 20.

$$\text{Efficiency} = \frac{(T_t)_{info}}{(T_t)_{info} + (T_{proc.})_{info} + 2 \times T_p + (T_t)_{ack}} \quad \ldots(i)$$

where,

$(T_t)_{info}$ = information frame transmission time

$(T_{proc.})_{info}$ = processing time of information frame = 0.25 ms

T_p = propagation delay = 0.75 ms

$(T_t)_{ask}$ = transmission time of acknowledge frame

$$(T_t)_{info} = \frac{\text{Information frame size}}{\text{Bandwidth}}$$

$$= \frac{2000 \times 8}{1 \times 10^6} = 16 \text{ ms}$$

$$(T_t)_{ask} = \frac{\text{Acknowledgement frame size}}{\text{Bandwidth}}$$

$$= \frac{20\times 8}{1\times 10^6} = 0.16 \text{ ms}$$

Now, putting all these values in equation (i),

$$\text{Efficiency} = \frac{16}{16+2\times 0.75\times 0.25+0.16}$$

$$= 89.34\ \%$$

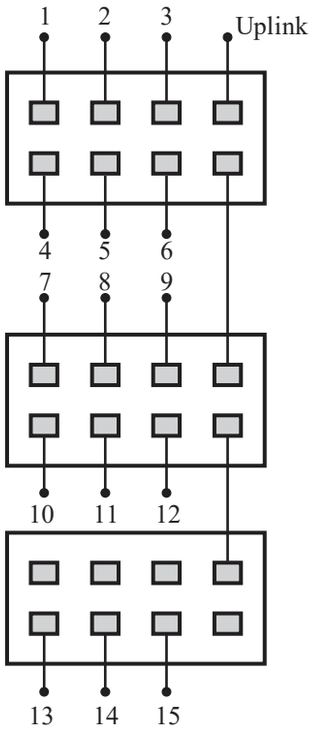
Hence, the correct answer is 89.34.

7 **3**

Given :

There are 15 machines connected in a LAN by using 8 port Ethernet switches,

Therefore,



So, from above figure the minimum number of switches needed is 3.

Hence, the correct answer is 3.

8 **(A), (B), (C)**

Consider each option one by one :

Option (A) : If the client was waiting to receive a packet, it may wait indefinitely.

It is correct, because client doesn't have a keep-alive timer, and the server after a reboot, forgets that any connection with the client existed.

Option (B) : The TCP server application on S can listen on P after reboot.

It is correct as once the current session will be terminated (due to reboot) and new connection can take place on the same port.

Option (C) : If the client sends a packet after the server reboot, it will receive a RST segment.

It is also correct as explained in option (A).

Option (D) : If the client sends a packet after the server reboot, it will receive a FIN segment.

It is false as the FIN is used to close connection.

Hence, the correct option is (A), (B) and (C).

❖❖❖❖

29 YEARS GATE SOLVED PAPERS

4 Years ESE Solved Papers

Exclusive With Synopsis

GATE 2022

UMESH DHANDE

ENGINEERING MATHEMATICS

EE CE CH EC ME IN PI

Topic-wise Previous 29 Years GATE Solutions

- Diligent solutions of GATE previous year questions (1993-2021)
- Multi method approach for a single problem to develop crystal clear concepts
- This book is also a value addition for ESE/PSUs/DRDO/ISRO
- Video Solution for Conspicuous Questions to Enhance problem solving Skills

GATE ACADEMY®
steps to success...