Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	What is collision resolution with open addressing?
((OPTION_A))	When collision happens, we create a new memory location outside of the existing table, and use a chain to link to the new memory location.
((OPTION_B))	When collision happens, we enlarge the hash table.
((OPTION_C))	When collision happens, we look for an unoccupied memory location in the existing table
((OPTION_D))	Use an extra table to collect all collided data
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Which of the following has a desired key is searched, starting itself from hash address, sequentially in a table?
((OPTION_A))	Linear Probing
((OPTION_B))	Quadratic Probing
((OPTION_C))	Chaining
((OPTION_D))	Reverse Probing
((CORRECT_CHOICE))	A

Unit - 3

(A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Folding is a method of generating
((OPTION_A))	A hash function
((OPTION_B))	Index function for a triangular matrix
((OPTION_C))	Linear probing
((OPTION_D))	Chaining.
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	When key values are reals a similar data representation might be produced by using a hashing function with
((OPTION_A))	Mod

Unit - 3

((OPTION_B))	Div
((OPTION_C))	Trunc
((OPTION_D))	Log N
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	An advantage of chained hash table over the open addressing scheme is
((OPTION_A))	Worst case complexity of serach operation os less
((OPTION_B))	Space used is less
((OPTION_C))	Deletion is easier
((OPTION_D))	None of these
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS))	1

Unit - 3

(1/2/3...)	
((QUESTION))	The average search time of hashing with linear probing will be less if the load factor
((OPTION_A))	Is far less than one
((OPTION_B))	Equals one
((OPTION_C))	Is far greater than one
((OPTION_D))	None of these
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A hash table has space for 100 records. What is the probability of collision before the table is 10% full?
((OPTION_A))	0.45
((OPTION_B))	0.5
((OPTION_C))	0.3
((OPTION_D))	0.34 (approximately)
((CORRECT_CHOICE))	A

Unit - 3

(A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Given a hash table T with 25 slots that stores 2000 elements, the load factor α for T is
((OPTION_A))	80
((OPTION_B))	0.0125
((OPTION_C))	8000
((OPTION_D))	1.25
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Which of the following statement(s) is TRUE? A hash function takes a message of arbitrary length and generates a fixed length code. A hash function takes a message of fixed length and generates a code of

Unit - 3

	variable length. A hash function may give the same hash value for distinct messages.
((OPTION_A))	1 only
((OPTION_B))	2 and 3 only
((OPTION_C))	1 and 3 only
((OPTION_D))	2 only
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Key density of a hash table is ratio of
((OPTION_A))	n/T
((OPTION_B))	T/n
((OPTION_C))	nT
((OPTION_D))	1/nT
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	In hash table, a record is stored using
((OPTION_A))	Index
((OPTION_B))	Key
((OPTION_C))	Function
((OPTION_D))	Table
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Most commonly used method in constructing hash function is
((OPTION_A))	Addition
((OPTION_B))	Subtraction
((OPTION_C))	Multiplication
((OPTION_D))	Division
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATI	

Unit - 3

ON)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	What is collision resolution with open addressing?
((OPTION_A))	When collision happens, we create a new memory location outside of the existing table, and use a chain to link to the new memory location.
((OPTION_B))	When collision happens, we enlarge the hash table.
((OPTION_C))	When collision happens, we look for an unoccupied memory location in the existing table
((OPTION_D))	Use an extra table to collect all collided data
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In _____ technique, instead of hashing function value as location we use it as an index into an array of pointers. Each pointer access a chain that holds the element having same location

Unit - 3

((OPTION_A))	Linear Probing
((OPTION_B))	Rehashing
((OPTION_C))	Chaining
((OPTION_D))	Bucket Hashing
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Each position of the hash table
((OPTION_A))	Bucket
((OPTION_B))	Slot
((OPTION_C))	Hashing
((OPTION_D))	Table
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	The mapping between an item and the slot where that item belongs in the hash table is called the
((OPTION_A))	Hash Function
((OPTION_B))	Hash Table
((OPTION_C))	Overflow
((OPTION_D))	Collision
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Load factor is ratio of
((OPTION_A))	Number of items in hash table to the hash table size
((OPTION_B))	Hash table size to number of items in hash table
((OPTION_C))	Slot to hash table size
((OPTION_D))	Hash table size to slot
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATI	

Unit - 3

ON)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Any function that can be used to map a data set of an arbitrary size to a data set of a fixed size which falls into the hash table is called
((OPTION_A))	Linear probing
((OPTION_B))	Open addressing
((OPTION_C))	Hashing
((OPTION_D))	Quadratic probing
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A data structure that associates keys (names) with values (attributes) is called
((OPTION_A))	Linear probing
((OPTION_B))	Collision

Unit - 3

((OPTION_C))	Hashing
((OPTION_D))	Hash Table
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	What is the hash key of 954 if hash function is x%3?
((OPTION_A))	0
((OPTION_B))	3
((OPTION_C))	4
((OPTION_D))	318
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1

Unit - 3

((QUESTION))	What is the hash key of 7564 if hash function is x%5?
((OPTION_A))	3
((OPTION_B))	4
((OPTION_C))	2
((OPTION_D))	1
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	What is the hash key of 7564 if hash function is x%10?
((OPTION_A))	4
((OPTION_B))	3
((OPTION_C))	2
((OPTION_D))	1
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION))	

Unit - 3

(OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A hash table partitioned into
((OPTION_A))	Slots
((OPTION_B))	Buckets
((OPTION_C))	Table
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In hash table if slot, s = 1 , each bucket can hold
((OPTION_A))	Two pairs
((OPTION_B))	One or more pairs
((OPTION_C))	Exactly one pair
((OPTION_D))	No pairs
((CORRECT_C	C

Unit - 3

HOICE)) (A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In hash table, Overflow occurred when
((OPTION_A))	Bucket is full
((OPTION_B))	Keys are full
((OPTION_C))	Only A
((OPTION_D))	Both A and B
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Assuming that the hash function for a table works well, and the size of the hash table is reasonably large compared to the number of items in the

Unit - 3

	table, the expected (average) time needed to find an item in a hash table containing n items is
((OPTION_A))	O(1)
((OPTION_B))	O(log n)
((OPTION_C))	O(nlog n)
((OPTION_D))	O(n)
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Suppose you place m items in a hash table with an array size of s. What is the correct formula for the load factor?
((OPTION_A))	s + m
((OPTION_B))	m – s
((OPTION_C))	m * s
((OPTION_D))	m / s
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION))	

Unit - 3

(OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	The searching technique that takes O (1) time to find a data is
((OPTION_A))	Linear Search
((OPTION_B))	Hashing
((OPTION_C))	Binary Search
((OPTION_D))	Tree Search
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	The goal of hashing is to produce a search that takes
((OPTION_A))	O(1) time
((OPTION_B))	O(log n) time
((OPTION_C))	$O(n^2)$ time
((OPTION_D))	O(n log n) time
((CORRECT_C	

Unit - 3

HOICE)) (A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	An integer key x is divided by the table size m and the remainder is taken as the hash value. Which type of hash function is this?
((OPTION_A))	Division method
((OPTION_B))	Folding method
((OPTION_C))	Mid square method
((OPTION_D))	Digit analysis
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A key is multiplied by itself and the hash value is obtained by selecting an appropriate number of digits from the middle of the square. The same

Unit - 3

	positions in the square must be used for all keys. Which hash function is this?
((OPTION_A))	Division method
((OPTION_B))	Folding method
((OPTION_C))	Mid square method
((OPTION_D))	Digit analysis
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A key is broken into several parts. Each part has the same length as that of the required address except the last part. The parts are added together, ignoring the last carry, we obtain the hash address for key K. Which hash function is this?
((OPTION_A))	Division method
((OPTION_B))	Folding method
((OPTION_C))	Mid square method
((OPTION_D))	Digit analysis
((CORRECT_CHOICE)) (A/B/C/D)	B

Unit - 3

((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Consider a key 7654321 is transformed into the address 1247 by selecting digits in position 1,2,4 and 7 then by reversing their order. Which method is used to calculate hash function
((OPTION_A))	Division method
((OPTION_B))	Folding method
((OPTION_C))	Mid square method
((OPTION_D))	Digit analysis
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A _____ is simply a mathematical formula that manipulates the key in some form to compute the index for this key in the hash table
((OPTION_A))	Hash function

Unit - 3

((OPTION_B))	Collision
((OPTION_C))	Only A
((OPTION_D))	Both A and B
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	The process of mapping keys to appropriate slots in a hash table is known as
((OPTION_A))	Tree Search
((OPTION_B))	Binary search
((OPTION_C))	Hashing
((OPTION_D))	Linear Search
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	A _____ is a data structure in which the location of a data item is determined directly as a function of data item itself rather than by a sequence of comparison.
((OPTION_A))	Tress
((OPTION_B))	Linked List
((OPTION_C))	Heap
((OPTION_D))	Hash table
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A _____ is simply a mathematical formula that manipulates the key in some form to compute the index for this key in the hash table
((OPTION_A))	Search Function
((OPTION_B))	Hash Function
((OPTION_C))	Division method
((OPTION_D))	Folding method
((CORRECT_CHOICE)) (A/B/C/D)	B

Unit - 3

((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Rehashing is a technique in which
((OPTION_A))	Second hash function is applied
((OPTION_B))	Chains are used
((OPTION_C))	Table size is changed
((OPTION_D))	Buckets are changed
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	_____maps the set of actual key values to the table without any collisions.
((OPTION_A))	Quadratic probing
((OPTION_B))	Perfect hashing
((OPTION_C))	Hashing

Unit - 3

((OPTION_D))	Minimal perfect hashing
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	
((OPTION_A))	Quadratic probing
((OPTION_B))	Perfect hashing
((OPTION_C))	Hashing
((OPTION_D))	Minimal perfect hashing
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	When the hash function produces the same address for distinct keys, it is called
((OPTION_A))	Collision

Unit - 3

((OPTION_B))	Indexing
((OPTION_C))	Probing
((OPTION_D))	Synonym
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Refreshing is a technique in which
((OPTION_A))	Second hash function is applied
((OPTION_B))	Table size is changed
((OPTION_C))	Chains are used
((OPTION_D))	Probes are used
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1

Unit - 3

((QUESTION))	A good hash function must have
((OPTION_A))	Minimum collisions
((OPTION_B))	Easy and quick to compute
((OPTION_C))	Distribute the keys evenly over hash table
((OPTION_D))	All of above
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	_____ maintains the record in order of hash field values
((OPTION_A))	Folding method
((OPTION_B))	Division method
((OPTION_C))	Indexed hashing
((OPTION_D))	Rehashing
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	In _____ arithmetic and logical function is applied on different field value to calculate hash address
((OPTION_A))	Open hashing
((OPTION_B))	Folding
((OPTION_C))	Division method
((OPTION_D))	Chaining
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	_____ is required when hash table is completely full.
((OPTION_A))	Rehashing
((OPTION_B))	Quadratic probing
((OPTION_C))	Overflow
((OPTION_D))	Probing
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATI	

Unit - 3

ON)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	_____ is required with quadratic probing when table is filled half.
((OPTION_A))	Rehashing
((OPTION_B))	Quadratic probing
((OPTION_C))	Overflow
((OPTION_D))	Probing
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	_____ is required when insertion is failed due to overflowing.
((OPTION_A))	Rehashing
((OPTION_B))	Quadratic probing
((OPTION_C))	Overflow
((OPTION_D))	Probing

Unit - 3

((CORRECT_CHOICE)) (A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Contents of old hash table to new hash table are transferred during
((OPTION_A))	Rehashing
((OPTION_B))	Quadratic probing
((OPTION_C))	Chaining
((OPTION_D))	Linear Probing
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In _____ space gets doubled
((OPTION_A))	Rehashing

Unit - 3

((OPTION_B))	Quadratic probing
((OPTION_C))	Chaining
((OPTION_D))	Linear Probing
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In browser program for caching the web pages _____ is used
((OPTION_A))	Linked list
((OPTION_B))	Rehashing
((OPTION_C))	Arrays or buffers
((OPTION_D))	Stack
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	The loading factor α is less than in following hashing techniques

Unit - 3

((OPTION_A))	Linear probing
((OPTION_B))	Quadratic probing
((OPTION_C))	Double hashing
((OPTION_D))	All of above
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In open addressing with load factor α, assume uniform hashing, the expected number of probes in an unsuccessful search is at the most
((OPTION_A))	1 - α
((OPTION_B))	A
((OPTION_C))	1/ (1- α)
((OPTION_D))	1/ α
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	What will be the loading factor of half-filled hash table of any size?
((OPTION_A))	0.25
((OPTION_B))	0.5
((OPTION_C))	0.75
((OPTION_D))	1.25
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	The hash table of size 8 contains exactly 5 records in it. The loading factor is
((OPTION_A))	0.5
((OPTION_B))	0.4
((OPTION_C))	0.6
((OPTION_D))	0.45
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATI	

Unit - 3

ON)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Process of finding some other position when hash address is occupied is classified as
((OPTION_A))	collision resolution
((OPTION_B))	address space resolution
((OPTION_C))	multiple hashing resolution
((OPTION_D))	chaining resolution
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Hashing technique which allocates fixed number of buckets is classified as
((OPTION_A))	dynamic hashing
((OPTION_B))	static hashing
((OPTION_C))	external hashing
((OPTION_D))	internal hashing
((CORRECT_C	C

Unit - 3

HOICE)) (A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Hashing technique which allows increase or decrease in number of buckets without a need of directory is classified as
((OPTION_A))	global depth hashing
((OPTION_B))	linear hashing
((OPTION_C))	relative hashing
((OPTION_D))	local depth hashing
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In linear hashing, formula used to calculate number of records if blocking factor, loading factor and file buckets are known is as
((OPTION_A))	r =l + bfr + N

Unit - 3

((OPTION_B))	r =l - bfr - N
((OPTION_C))	r =l + bfr - N
((OPTION_D))	r =l * bfr * N
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	If global depth is more than local depth then operation must be performed in directory array is
((OPTION_A))	adding
((OPTION_B))	subtracting
((OPTION_C))	halving
((OPTION_D))	doubling
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	In linear hashing, formula of file load factor is
((OPTION_A))	l = r⁄ (bfr * N)
((OPTION_B))	l = r⁄ (bfr + N)
((OPTION_C))	l = r⁄ (bfr - N)
((OPTION_D))	l = r⁄ (bfr * 2N)
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((OPTION_A))	What causes a collision?
((OPTION_B))	The program you are running crashes
((OPTION_C))	There are too many hash keys in the array
((OPTION_D))	Two hash keys are the same
((CORRECT_CHOICE)) (A/B/C/D)	The program is out of memory
((EXPLANATION)) (OPTIONAL)	C

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	What are the three types of collision solutions?
((OPTION_A))	Overflow, underflow and undertow
((OPTION_B))	Chaining, overflow and probing
((OPTION_C))	Probing, underflow and chaining
((OPTION_D))	Noflow, fastflow and chaining
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	What does hashing improve?
((OPTION_A))	Speed
((OPTION_B))	Eye-stress
((OPTION_C))	Mood
((OPTION_D))	Hard drive space
((CORRECT_CHOICE)) (A/B/C/D)	A

Unit - 3

((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	How is a hash key computed?
((OPTION_A))	Subtraction
((OPTION_B))	Modulo division
((OPTION_C))	Random number generation
((OPTION_D))	All of the above
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Consider a hash function that distributes keys uniformly. The hash table size is 20. After hashing of how many keys will the probability that any new key hashed collides with an existing one exceed 0.5
((OPTION_A))	5
((OPTION_B))	6

Unit - 3

((OPTION_C))	7
((OPTION_D))	10
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In universal hashing refers to selecting a hash function at random from a family of hash functions with a certain -------- property
((OPTION_A))	mathematical
((OPTION_B))	random
((OPTION_C))	specific
((OPTION_D))	None of the above
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	Linear hashing is example of ------- hash functions
((OPTION_A))	dynamic
((OPTION_B))	static
((OPTION_C))	Semi-static
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A good hash procedure must be -----
((OPTION_A))	static
((OPTION_B))	non-deterministic
((OPTION_C))	deterministic
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	In which of the following method, the key is the address without any algorithmic manipulation.
((OPTION_A))	Mid-Square
((OPTION_B))	Modulo-Division
((OPTION_C))	Folding
((OPTION_D))	Direct hashing
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In which of the following method, the key value is divided into parts whose size matches the size of the required address.
((OPTION_A))	Mid-Square
((OPTION_B))	Modulo-Division
((OPTION_C))	Fold shift
((OPTION_D))	Fold Boundary
((CORRECT_C	C

Unit - 3

HOICE)) (A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	In which of the following method, the left and right numbers are folded on a fixed boundary between them and the center number.
((OPTION_A))	Mid-Square
((OPTION_B))	Modulo-Division
((OPTION_C))	Fold shift
((OPTION_D))	Fold Boundary
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	The load Factor alpha of a hash table of size M with N occupied entries is defined by
((OPTION_A))	N/M
((OPTION_B))	M/N

Unit - 3

((OPTION_C))	M*N
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	------- clustering occurs when data become clustered around a home address.
((OPTION_A))	Primary
((OPTION_B))	Secondary
((OPTION_C))	Complex
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS))	1

Unit - 3

(1/2/3...)	
((QUESTION))	------- clustering occurs when data become grouped along a collision path throughout the list.
((OPTION_A))	Primary
((OPTION_B))	Secondary
((OPTION_C))	Complex
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	One way of resolving collisions is to maintain M linked lists, one for each possible address in the hash table is known as
((OPTION_A))	Linear Probe
((OPTION_B))	Quadratic Probe
((OPTION_C))	Double Hashing
((OPTION_D))	Chaining
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION))	

Unit - 3

(OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A skip list is built in
((OPTION_A))	0 layer
((OPTION_B))	1 layer
((OPTION_C))	Multiple layers
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	The bottom layer in Skip List is an ordinary ordered ----
((OPTION_A))	Array
((OPTION_B))	Linked list
((OPTION_C))	It depends
((OPTION_D))	None
((CORRECT_C	B

Unit - 3

HOICE)) (A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A skiplist is capable of ------ insertion
((OPTION_A))	O(log n)
((OPTION_B))	O(n)
((OPTION_C))	O(n*log n)
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A skiplist is capable of ------ removal of values from a sorted sequence
((OPTION_A))	O(log n)

Unit - 3

((OPTION_B))	O(n)
((OPTION_C))	O(n*log n)
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	A skiplist is capable of ------ lookups of values at a given position in the sequence.
((OPTION_A))	O(log n)
((OPTION_B))	O(n)
((OPTION_C))	O(n*log n)
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1

Unit - 3

((QUESTION))	In skiplist with a minor modification the speed of random access indexed lookups can be improved to
((OPTION_A))	O(log n)
((OPTION_B))	O(n)
((OPTION_C))	O(n*log n)
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Which it true? i. Skip list algorithms have the same asymptotic expected time bounds as balanced trees. ii. They are simpler, faster and use less space.
((OPTION_A))	i only
((OPTION_B))	ii only
((OPTION_C))	Both
((OPTION_D))	None
((CORRECT_CHOICE))	C

Unit - 3

(A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Skiplist find the k-th largest element in the set, in ---- time
((OPTION_A))	O(log n)
((OPTION_B))	O(n)
((OPTION_C))	O(n*log n)
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Dictionary can be implemented using
((OPTION_A))	Hashtable
((OPTION_B))	Singly Linked List
((OPTION_C))	Binary Search Tree
((OPTION_D))	All of the above
((CORRECT_C	D

Unit - 3

HOICE)) (A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Applications of Dictionary can be
((OPTION_A))	address book
((OPTION_B))	credit card authorization
((OPTION_C))	mapping host names
((OPTION_D))	All of the above
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	The expected running time of all the dictionary ADT operations in a hash table is

Unit - 3

((OPTION_A))	O(1)
((OPTION_B))	O(log n)
((OPTION_C))	O(n)
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Application of hash table is
((OPTION_A))	small databases
((OPTION_B))	compilers
((OPTION_C))	browser caches
((OPTION_D))	All of the above
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1

Unit - 3

((QUESTION))	Assuming that the keys are random numbers, it can be shown that the expected number of probes for an insertion with open addressing is
((OPTION_A))	1/ α
((OPTION_B))	1 / (1−α)
((OPTION_C))	α
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Which of the following methods are for ordered dictionaries?
((OPTION_A))	closestKeyBefore
((OPTION_B))	closestElemBefore
((OPTION_C))	closestKeyAfter
((OPTION_D))	All of these
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	1
((QUESTION))	Which of the following methods are for unordered dictionaries?
((OPTION_A))	closestKeyBefore(key)
((OPTION_B))	closestElemBefore(key)
((OPTION_C))	closestKeyAfter(key)
((OPTION_D))	None of these
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	More efficient implementations of an ordered dictionary are
((OPTION_A))	BST
((OPTION_B))	AVL trees
((OPTION_C))	Both
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	C

Unit - 3

((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	1
((QUESTION))	Implementations of Dictionary ADT can be done using
((OPTION_A))	Log File
((OPTION_B))	Ordered Dictionary
((OPTION_C))	Hash Table
((OPTION_D))	All of these
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	A hash table with 10 buckets with one slot per bucket is depicted. The symbols, S1 to S7 are initially emerged using a hashing function with linear probing. Maximum number of comparisons needed in searching an item that is not present is
((OPTION_A))	6
((OPTION_B))	5
((OPTION_C))	4

Unit - 3

((OPTION_D))	3
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	A hash function f defined as f(key)=key mod 7,with linear probling, insert the keys 37,38,72,48,98,11,56, into a table indexed from 11 will be stored in the location
((OPTION_A))	3
((OPTION_B))	4
((OPTION_C))	5
((OPTION_D))	6
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS))	2

Unit - 3

(1/2/3...)	
((QUESTION))	A hash function randomly distributes records one by one in a space that can hold x number of records. The probability that the m^{th} record is the first record to result in collision is
((OPTION_A))	$(x-1)\ (x-2)...(x-(m-2))(m-1)/x^{m-1}$
((OPTION_B))	$(x-1)\ (x-2)...(x-(m-1))(m-1)/x^{m-1}$
((OPTION_C))	$(x-1)\ (x-2)...(x-(m-2))(m-1)/x^{m}$
((OPTION_D))	$(x-1)\ (x-2)...(x-(m-1))(m-1)/x^{m}$
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	A _____ is a data structure in which the location of a data item is determined directly as a function of data item itself rather than by a sequence of comparison. Under ideal condition, the time required to locate a data item in a hash table is 0(1) ie. It is constant and DOES not depend on the number of data items stored. When the set of K of keys stored is much smaller than the universe U of all possible keys, a hash table require much less storage spare than a direct address table.
((OPTION_A))	Hash table
((OPTION_B))	Symbol Table
((OPTION_C))	Bucket Table

Unit - 3

((OPTION_D))	Dynamic Table
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	The key space is much larger than the address space, therefore, many keys are mapped to the same address. Suppose that two keys K1 and K2 map to the same address. When the record with key K1 is entered, it is inserted at the hashed address, but when another record with key K2 is entered, it is a dilemma where to insert as a record with key K1 is already there. This situation is called a _____.
((OPTION_A))	Overflow
((OPTION_B))	Collision
((OPTION_C))	Division
((OPTION_D))	Digit Analysis
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	2
((QUESTION))	Which of the following statement(s) is TRUE about static hashing? 1. Numbers of buckets are fixed 2. Implementation is complex. 3. Space overhead is more
((OPTION_A))	1 only
((OPTION_B))	2 and 3 only
((OPTION_C))	1 and 3 only
((OPTION_D))	2 only
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Which of the following statement(s) is TRUE about dynamic hashing? 1. Numbers of buckets are not fixed 2. Space overhead is more 3. Implementation is complex.
((OPTION_A))	1 only
((OPTION_B))	2 and 3 only

Unit - 3

((OPTION_C))	1 and 3 only
((OPTION_D))	2 only
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Consider a hash table with h = 26 buckets and s = 2 slots. Assume there are n = 10 distinct keys. What will the loading density for this table?
((OPTION_A))	0.19
((OPTION_B))	0.38
((OPTION_C))	0.25
((OPTION_D))	0.01
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	2
((QUESTION))	There are several factors that affect the efficiency of lookup operations in a hash table. Which of the following is not one of those factors? A. Number of elements stored in the hash table B. Size of elements stored in the hash table C. Number of buckets in the hash table D. Quality of the hash function
((OPTION_A))	Only B
((OPTION_B))	B and C only
((OPTION_C))	A and B only
((OPTION_D))	All A,B,C,D
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Mark all properties that are TRUE for a hashtable with n elements? 1. it is possible to have different keys being hashed to the same position in the array 2. an ideal hash table using array doubling has average-case time complexity of O(1) for lookups 3. can be used to sort an array of n real numbers with average-case

Unit - 3

	time complexity O(n) 4. an ideal hash table using array doubling has worst-case time complexity of O(1) for every insert operation
((OPTION_A))	1and 2 only
((OPTION_B))	2and 3 only
((OPTION_C))	2 and 4 only
((OPTION_D))	1 and 4 only
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	If hash function is remainder to division, then clustering is more likely to occur if storage space is divided into 40 sectors rather than 41, this conclusion is
((OPTION_A))	More likely to be false
((OPTION_B))	More likely to be true
((OPTION_C))	Is always false
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	B

Unit - 3

((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Which of following statement is true? 1. As number of entries in the hash table increases, the number of collisions increases 2. Number of buckets are fixed in static hashing 3. Implementation of dynamic hashing is easy. 4. A hash function takes a message of arbitrary length and generates a fixed length code.
((OPTION_A))	1 and 3 only
((OPTION_B))	2 and 3 only
((OPTION_C))	3 and 4 only
((OPTION_D))	1, 2, and 4 only
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS))	2

Unit - 3

(1/2/3...)	
((QUESTION))	Adding the objects to the hash table for list of elements which are already sorted give following result
((OPTION_A))	Placing the elements in the table becomes time efficient
((OPTION_B))	Placing the elements in the table becomes space efficient
((OPTION_C))	There is no need to resize table
((OPTION_D))	No effect
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	If the hash table is half full then expected number of probes in it with unsuccessful search is
((OPTION_A))	0.5
((OPTION_B))	0.75
((OPTION_C))	2
((OPTION_D))	1
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATI	

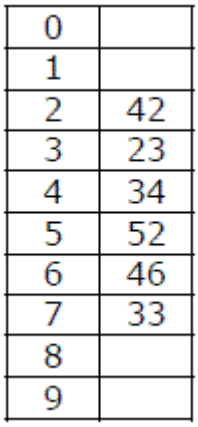
Unit - 3

ON)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	What is the hash key value of 'Donaldson' if you assign each letter its corresponding number in the alphabet (ie f = 6) and if you use 9 as the divisor?
((OPTION_A))	12
((OPTION_B))	3
((OPTION_C))	7
((OPTION_D))	None of the above
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing.

Unit - 3

	 After inserting 6 values into an empty hash table, the table is as shown below. Which one of the following choices gives a possible order in which the key values could have been inserted in the table?
((OPTION_A))	46, 42, 34, 52, 23, 33
((OPTION_B))	34, 42, 23, 52, 33, 46
((OPTION_C))	46, 34, 42, 23, 52, 33
((OPTION_D))	42, 46, 33, 23, 34, 52
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is the resultant hash table?

Unit - 3

((OPTION_A))	
((OPTION_B))	
((OPTION_C))	
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	2
((QUESTION))	Consider a hash table of size seven, with starting index zero, and a hash function (3x + 4)mod7. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing? Note that '_' denotes an empty location in the table.
((OPTION_A))	8, _, _, _, _, _, 10
((OPTION_B))	1, 8, 10, _, _, _, 3
((OPTION_C))	1, _, _, _, _, _,3
((OPTION_D))	1, 10, 8, _, _, _, 3
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true? i. 9679, 1989, 4199 hash to the same value ii. 1471, 6171 has to the same value iii. All elements hash to the same value iv. Each element hashes to a different value
((OPTION_A))	i only
((OPTION_B))	ii only
((OPTION_C))	i and ii only

Unit - 3

((OPTION_D))	iii or iv
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 3 slots are unfilled after the first 3 insertions?
((OPTION_A))	$(97 \times 97 \times 97)/100^3$
((OPTION_B))	$(99 \times 98 \times 97)/100^3$
((OPTION_C))	$(96 \times 95 \times 94)/100^3$
((OPTION_D))	None of the above
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2

Unit - 3

((QUESTION))	Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2020?
((OPTION_A))	$h(i) = i^2 \bmod 10$
((OPTION_B))	$h(i) = i^3 \bmod 10$
((OPTION_C))	Both
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Given a hash table T with 25 slots that stores 2000 elements, the load factor α for T is
((OPTION_A))	80
((OPTION_B))	20.5
((OPTION_C))	8000
((OPTION_D))	1.25
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

Unit - 3

((MARKS)) (1/2/3...)	2
((QUESTION))	Consider a hash table with nn buckets, where external (overflow) chaining is used to resolve collision. The hash function is such that the probability that a key value is hashed to a particular bucket is 1/n. The hash table is initially empty and k distinct values are inserted in the table. What is the probability that bucket number 1 is empty after the k^{th} insertion?
((OPTION_A))	n
((OPTION_B))	$(n-1)^k$
((OPTION_C))	$((n-1)/n)^k$
((OPTION_D))	$((n-1)/k)^n$
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Extendible hashing is a type of hash system which treats a hash as a bit string, and uses a ------ for bucket lookup
((OPTION_A))	Trie
((OPTION_B))	HashMap

Unit - 3

((OPTION_C))	Both
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	A
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	In Extendible hashing re-hashing is an ------ operation
((OPTION_A))	Parallel
((OPTION_B))	Incremental
((OPTION_C))	Semi-parallel
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Which of the following statment is true? i. Linear hashing allows for the expansion of the hash table one slot at a

Unit - 3

	time. ii. The frequent single slot expansion can very effectively control the length of the collision chain.
((OPTION_A))	i only
((OPTION_B))	ii only
((OPTION_C))	Both
((OPTION_D))	None
((CORRECT_CHOICE)) (A/B/C/D)	C
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Which of the following statment is true? i. In universal hashing refers to selecting a hash function at random from a family of hash functions with a certain mathematical property. ii. This guarantees a low number of collisions in expectation, even if the data is chosen by an adversary.
((OPTION_A))	i only
((OPTION_B))	ii only
((OPTION_C))	Both
((OPTION_D))	None
((CORRECT_C	C

Unit - 3

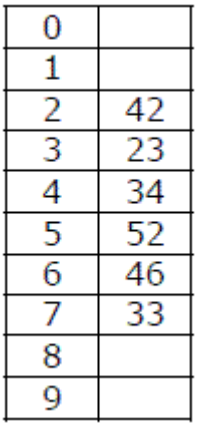
HOICE)) (A/B/C/D)	
((EXPLANATION)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	Which of the following statment is true? i. A hash function is any function that can be used to map data of fixed size to data of arbitrary size. ii. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes. iii. One use is a data structure called a hash table, widely used in computer software for rapid data lookup. iv. Hash functions accelerate table or database lookup by detecting duplicated records in a large file.
((OPTION_A))	i, ii, iii
((OPTION_B))	ii, iii, iv
((OPTION_C))	iii, iv
((OPTION_D))	All of these
((CORRECT_CHOICE)) (A/B/C/D)	B
((EXPLANATI	

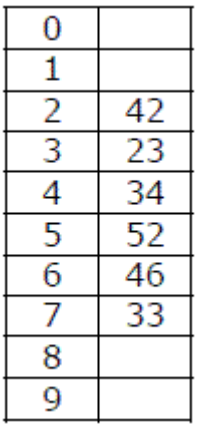
Unit - 3

ON)) (OPTIONAL)	

((MARKS)) (1/2/3...)	2
((QUESTION))	A search for a target element in Skip List begins at the head element in the top list, and proceeds horizontally until the current element is ------ the target.
((OPTION_A))	less than
((OPTION_B))	less than or equal to
((OPTION_C))	greater than
((OPTION_D))	greater than or equal to
((CORRECT_CHOICE)) (A/B/C/D)	D
((EXPLANATION)) (OPTIONAL)	

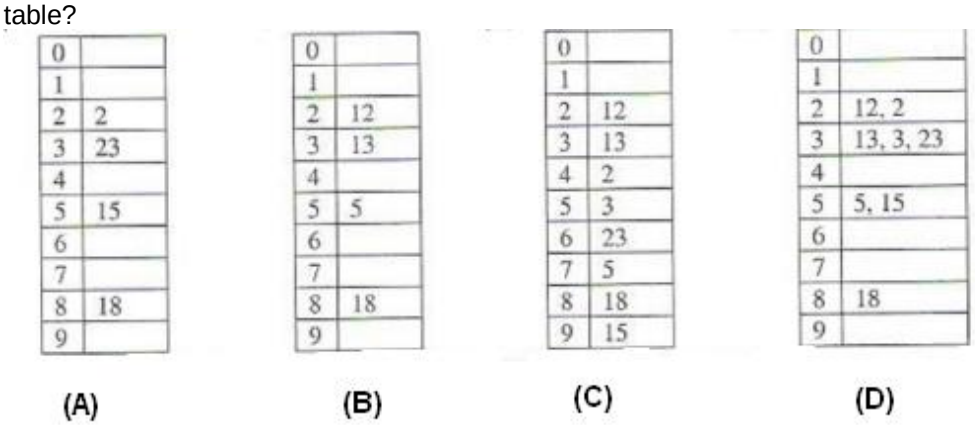
((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.  Which one of the following choices gives a possible order in which the key values could have been inserted in the table?
((OPTION_A)) THIS IS MANDATORY OPTION	46,42,34,52,23,33
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	34,42,23,52,33,46
((OPTION_C)) This is optional	46,34,42,23,52,33
((OPTION_D)) This is optional	42,46,33,23,34,52
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or	C

E	
((EXPLANATION)) This is also optional	The sequence (A) doesn't create the hash table as the element 52 appears before 23 in this sequence. The sequence (B) doesn't create the hash table as the element 33 appears before 46 in this sequence. The sequence (C) creates the hash table as 42, 23 and 34 appear before 52 and 33, and 46 appears before 33. The sequence (D) doesn't create the hash table as the element 33 appears before 23 in this sequence.

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	How many different insertion sequences of the key values using  the same hash function h(k)=k mod 10 and linear probing will result in the hash table shown above?
((OPTION_A)) THIS IS MANDATORY OPTION	10
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	20
((OPTION_C)) This is optional	30
((OPTION_D)) This is optional	40
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	In a valid insertion sequence, the elements 42, 23 and 34 must appear before 52 and 33, and 46 must appear before 33. Total number of different sequences = **3! x 5** = 30 In the above expression, **3!** is for elements 42, 23 and 34 as they can appear in any order, and **5** is for element 46 as it can appear at 5 different places

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is the resultant hash table?  (A) (B) (C) (D)
((OPTION_A)) THIS IS MANDATORY OPTION	A
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	B
((OPTION_C))	C

This is optional	
((OPTION_D)) This is optional	D
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	Open addressing, or closed hashing, is a method of collision resolution in hash tables. With this method a hash collision is resolved by probing, or searching through alternate locations in the array (the probe sequence) until either the target record is found, or an unused array slot is found, which indicates that there is no such key in the table. Well known probe sequences include: *linear probing* in which the interval between probes is fixed--often at 1. *quadratic probing* in which the interval between probes increases linearly (hence, the indices are described by a quadratic function). *double hashing* in which the interval between probes is fixed for each record but is computed by another hash function.

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Consider a hash table of size seven, with starting index zero, and a hash function (3x + 4)mod7. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing? Note that '_' denotes an empty location in the table.
((OPTION_A)) THIS IS MANDATORY OPTION	8, _, _, _, _, _, 10
((OPTION_B))	1, 8, 10, _, _, _, 3

THIS IS ALSO MANDATORY OPTION	
((OPTION_C)) This is optional	1, _, _, _, _, _,3
((OPTION_D)) This is optional	1, 10, 8, _, _, _, 3
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	B
((EXPLANATION)) This is also optional	let us put values 1, 3, 8, 10 in the hash of size 7. Initially, hash table is empty - - - - - - - 0 1 2 3 4 5 6 The value of function (3x + 4)mod 7 for 1 is 0, so let us put the value at 0 1 - - - - - - 0 1 2 3 4 5 6 The value of function (3x + 4)mod 7 for 3 is 6, so let us put the value at 6 1 - - - - - 3 0 1 2 3 4 5 6 The value of function (3x + 4)mod 7 for 8 is 0, but 0 is already occupied, let us put the value(8) at next available space(1) 1 8 - - - - 3 0 1 2 3 4 5 6 The value of function (3x + 4)mod 7 for 10 is 6, but 6 is already occupied, let us put the value(10) at next available space(2) 1 8 10 - - - 3 0 1 2 3 4 5 6

((MARKS))	1

QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true? i. 9679, 1989, 4199 hash to the same value ii. 1471, 6171 has to the same value iii. All elements hash to the same value iv. Each element hashes to a different value
((OPTION_A)) THIS IS MANDATORY OPTION	I only
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	Ii only
((OPTION_C)) This is optional	I and ii only
((OPTION_D)) This is optional	Iii or iv
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	Hash function given is mod(10). 9679, 1989 and 4199 all these give same hash value i.e 9 1471 and 6171 give hash value 1

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 3 slots are unfilled after the first 3 insertions?
((OPTION_A)) THIS IS MANDATORY OPTION	$(97 \times 97 \times 97)/100^3$
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	$(99 \times 98 \times 97)/100^3$
((OPTION_C)) This is optional	$(97 \times 96 \times 95)/100^3$
((OPTION_D)) This is optional	$(97 \times 96 \times 95)/(3! \times 100^3)$
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	A

((EXPLANATION)) This is also optional	Simple Uniform hashing function is a hypothetical hashing function that evenly distributes items into the slots of a hash table. Moreover, each item to be hashed has an equal probability of being placed into a slot, regardless of the other elements already placed.
	`Probability that the first 3 slots are unfilled after the first 3 insertions =`
	`(probability that first item doesn't go in any of the first 3 slots)*`

	(probability that second item doesn't go in any of the first 3 slots)* (probability that third item doesn't go in any of the first 3 slots) = (97/100) * (97/100) * (97/100)

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2020?
((OPTION_A)) THIS IS MANDATORY OPTION	$h(i) = i^2 \bmod 10$
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	$h(i) = i^3 \bmod 10$
((OPTION_C)) This is optional	$h(i) = (11 * i^2) \bmod 10$
((OPTION_D)) This is optional	$h(i) = (12 * i) \bmod 10$
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	

((CORRECT_CHOICE)) Either A or B or C or D or E	B
((EXPLANATION)) This is also optional	Since mod 10 is used, the last digit matters. If you do cube all numbers from 0 to 9, you get following Number Cube Last Digit in Cube 0 0 0 1 1 1 2 8 8 3 27 7 4 64 4 5 125 5 6 216 6 7 343 3 8 512 2 9 729 9 Therefore all numbers from 0 to 2020 are equally divided in 10 buckets. If we make a table for square, we don't get equal distribution. In the following table. 1, 4, 6 and 9 are repeated, so these buckets would have more entries and buckets 2, 3, 7 and 8 would be empty. Number Square Last Digit in Cube 0 0 0 1 1 **1** 2 4 **4** 3 9 **9** 4 16 **6** 5 25 5 6 36 **6** 7 49 **9** 8 64 **4** 9 81 **1**

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER	Given a hash table T with 25 slots that stores 2000 elements, the load factor α for T is _____

CONTENT. QTN CAN HAVE IMAGES ALSO	
((OPTION_A)) THIS IS MANDATORY OPTION	80
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	0.0125
((OPTION_C)) This is optional	8000
((OPTION_D)) This is optional	1.25
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	A
((EXPLANATION)) This is also optional	load factor = (no. of elements) / (no. of table slots) = 2000/25 = 80

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER	Which of the following statement(s) is TRUE? 1. A hash function takes a message of arbitrary length and generates a fixed length code. 2. A hash function takes a message of fixed length and generates a code of

CONTENT. QTN CAN HAVE IMAGES ALSO	variable length. 3. A hash function may give the same hash value for distinct messages
((OPTION_A)) THIS IS MANDATORY OPTION	I only
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	Ii and iii only
((OPTION_C)) This is optional	I and iii only
((OPTION_D)) This is optional	Ii only
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	Hash function is defined as any function that can be used to map data of arbitrary size of data to a fixed size data.. The values returned by a hash function are called hash values, hash codes, digests, or simply hashes : Statement 1 is correct Yes, it is possible that a Hash Function maps a value to a same location in the memmory that's why collision occurs and we have different technique to handle this problem : Statement 3 is coorect. eg : we have hash function, h(x) = x mod 3 Acc to Statement 1, no matter what the value of 'x' is h(x) results in a fixed mapping location. Acc. to Statement 3, h(x) can result in same mapping mapping location for different value of 'x' e.g. if x = 4 or x = 7 , h(x) = 1 in both the cases, although collision occurs.

((MARKS)) QUESTION IS OF	1

HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Consider a hash function that distributes keys uniformly. The hash table size is 20. After hashing of how many keys will the probability that any new key hashed collides with an existing one exceed 0.5.
((OPTION_A)) THIS IS MANDATORY OPTION	5
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	6
((OPTION_C)) This is optional	7
((OPTION_D)) This is optional	10
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	D
((EXPLANATION)) This is also optional	For each entry probability of collision is 1/20 {as possible total spaces =20, and an entry will go into only 1 place} Say after inserting x values probability becomes ½ □ (1/20).x = ½ □ X=10

((MARKS))	1

QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Process of finding some other position when hash address is occupied is classified as
((OPTION_A)) THIS IS MANDATORY OPTION	collision resolution
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	address space resolution
((OPTION_C)) This is optional	multiple hashing resolution
((OPTION_D)) This is optional	chaining resolution
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	A
((EXPLANATION)) This is also optional	

((MARKS))	1

QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	We have a hash table of size 7 to store integer keys, with hash function h(x) = x mod 7. If we use linear probing and insert elements in the order 1, 15, 14, 3, 9, 5, 27,in which bucket 9 will be placed?
((OPTION_A)) THIS IS MANDATORY OPTION	3
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	4
((OPTION_C)) This is optional	2
((OPTION_D)) This is optional	5
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	B
((EXPLANATION)) This is also optional	Index key 0 14 1 1 2 15 3 3 4 9

	5 5 6 27 Collision occurred for 9 at 2 index, so we tried to find next available bucket i.e. 4

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	1. What causes a collision?
((OPTION_A)) THIS IS MANDATORY OPTION	The program you are running crashes
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	There are too many hash keys in the array
((OPTION_C)) This is optional	Two hash keys are the same
((OPTION_D)) This is optional	The program is out of memory
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or	C

E	
((EXPLANATION)) This is also optional	Collision is a situation when two keys demand for same location .i.e their hash address is same

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	What are the three types of collision solutions?
((OPTION_A)) THIS IS MANDATORY OPTION	Overflow, underflow and undertow
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	Chaining, rehashing and probing
((OPTION_C)) This is optional	Probing, underflow and chaining
((OPTION_D)) This is optional	None
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or	B

E	
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Hash function used to map several dictionary entry in hash table .Each position is called as
((OPTION_A)) THIS IS MANDATORY OPTION	Bucket
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	Probe
((OPTION_C)) This is optional	Synonym
((OPTION_D)) This is optional	None
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or	A

E	
((EXPLANATION)) This is also optional	Hash table is partitioned into different buckets

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	If hash function is hash(x)=x mod 10 then 21,101,121,151 are ---- of each other.
((OPTION_A)) THIS IS MANDATORY OPTION	Load factor
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	Synonym
((OPTION_C)) This is optional	Both a and b
((OPTION_D)) This is optional	None
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or	B

E	
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Which of below is type of hash function 1.Division 2.Bit extraction 3.Folding 4. Linear probing 5.Chaining 6.Rehashing
((OPTION_A)) THIS IS MANDATORY OPTION	1,2 and 3
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	All are correct
((OPTION_C)) This is optional	4,5 and 6
((OPTION_D)) This is optional	2,4 and 6
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	

((CORRECT_CHOICE)) Either A or B or C or D or E	A
((EXPLANATION)) This is also optional	Only first 3 are hash functions and rest are collision resolution strategy

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	For key 345678123 what will be index in fold shift and fold boundry respectively
((OPTION_A)) THIS IS MANDATORY OPTION	146 and 542
((OPTION_B)) THIS IS ALSO MANDATORY OPTION	641 and 245
((OPTION_C)) This is optional	542 and 146
((OPTION_D)) This is optional	678 and 876
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	

((CORRECT_CHOICE)) Either A or B or C or D or E	A
((EXPLANATION)) This is also optional	In fold shift, 345 678 123 345 +678 +123 --- 1146 discard 1 so index =146 In fold boundry, 543 +678 +321 --- 1542 discard 1 so index=542

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Which of the below is rules for choosing hash function? 1. easy to compute 2. Less no of collision 3. Should not depend on every bit of the key 4. It should produce keys which will get distributed un uniformly
((OPTION_A)) THIS IS MANDATORY OPTION	1 and 2
((OPTION_B)) THIS IS ALSO MANDATORY	All of the above

OPTION	
((OPTION_C)) This is optional	3 and 4
((OPTION_D)) This is optional	None of the above
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	A
((EXPLANATION)) This is also optional	It should depend on every bit of key and should distribute keys uniformly over the table so 3 and 4 are not correct

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Below is open hashing technique is collision resolution 1. Open addressing 2. Chaining 3. Rehashing 4. Double hashing
((OPTION_A)) THIS IS MANDATORY OPTION	1 only
((OPTION_B)) THIS IS ALSO MANDATORY	2 only

OPTION	
((OPTION_C)) This is optional	3 only
((OPTION_D)) This is optional	All of the above
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	B
((EXPLANATION)) This is also optional	In open hashing, collisions are stored outside the table And in closed hashing collisions are stored in same table at some other index

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	In quadratic probing with hash function hash(k)=k mod 10 , for keys 37,90,55,22,11,17,49,87 placed in hash table of size 10 ,what will be location of 87?
((OPTION_A)) THIS IS MANDATORY OPTION	7
((OPTION_B)) THIS IS ALSO MANDATORY	8

OPTION	
((OPTION_C)) This is optional	1
((OPTION_D)) This is optional	6
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	D
((EXPLANATION)) This is also optional	If collision situation , H(k)=(hash(k)+i^2) mod m

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	For keys : 37,90,45,22,17,49,55 placed in hash table of size 10, what will be location of 17?
((OPTION_A)) THIS IS MANDATORY OPTION	7
((OPTION_B)) THIS IS ALSO MANDATORY	4

OPTION	
((OPTION_C)) This is optional	1
((OPTION_D)) This is optional	2
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	In double hash in case of collision h2(key)=M-(key mod M) whatever 2nd hash index we generate, that many times need to move from home address here for 17 h1(17)=7 and h2(17)=4 i.e move 4 places from 7 so we get index 2 for storing 17

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	When we are using rehashing technique?
((OPTION_A)) THIS IS MANDATORY OPTION	When insertion fail due to overflow
((OPTION_B)) THIS IS ALSO	When table is almost filled and to avoid future failure in insertion operation

MANDATORY OPTION	
((OPTION_C)) This is optional	Both A and B
((OPTION_D)) This is optional	Only A
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	For keys 50, 700, 76, 85, 92, 73, 101 stored in hash table with hash function =key mod 7 and chaining is used for resolving collision. Which of the below is part of same list?
((OPTION_A)) THIS IS MANDATORY OPTION	50,85
((OPTION_B)) THIS IS ALSO	50,85,92

MANDATORY OPTION	
((OPTION_C)) This is optional	73,101
((OPTION_D)) This is optional	Both B and C
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	None of the above
((CORRECT_CHOICE)) Either A or B or C or D or E	D
((EXPLANATION)) This is also optional	All synonyms are part of same link list in chaining

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Dictionary consists of…… and is -----
((OPTION_A)) THIS IS MANDATORY OPTION	Pairs of <key,value>, of no specific order
((OPTION_B)) THIS IS ALSO	Keys,sorted

MANDATORY OPTION	
((OPTION_C)) This is optional	Pairs of<key,value>, sorted
((OPTION_D)) This is optional	None of the above
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Suppose you are given the following set of keys to insert into a hash table that holds exactly 11 values: 113 , 117 , 97 , 100 , 114 , 108 , 116 , 105 , 99 Which of the following best demonstrates the contents of the has table after all the keys have been inserted using linear probing?
((OPTION_A)) THIS IS MANDATORY OPTION	100, __, __, 113, 114, 105, 116, 117, 97, 108, 99
((OPTION_B)) THIS IS ALSO	99, 100, __, 113, 114, __, 116, 117, 105, 97, 108

MANDATORY OPTION	
((OPTION_C)) This is optional	100, 113, 117, 97, 14, 108, 116, 105, 99, __, __
((OPTION_D)) This is optional	117, 114, 108, 116, 105, 99, __, __, 97, 100, 113
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	B
((EXPLANATION)) This is also optional	Using modulo 11 arithmetic and linear probing gives these values

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	If total no of keys are 20 and hash table size is 40 then average no of comparisons of successful search in separate chaining is
((OPTION_A)) THIS IS MANDATORY OPTION	1.25
((OPTION_B)) THIS IS ALSO	0.5

MANDATORY OPTION	
((OPTION_C)) This is optional	0.25
((OPTION_D)) This is optional	2
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	A
((EXPLANATION)) This is also optional	Sn=1+(α/2) in chaining

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	What is average time complexity of searching in skip list?
((OPTION_A)) THIS IS MANDATORY OPTION	O(n)
((OPTION_B)) THIS IS ALSO	O(nlogn)

MANDATORY OPTION	
((OPTION_C)) This is optional	O(n/2)
((OPTION_D)) This is optional	O(log n)
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	D
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	What is true in extensible hashing?
((OPTION_A)) THIS IS MANDATORY OPTION	Hash function uses key directly
((OPTION_B)) THIS IS ALSO	Hash function takes Binary representation

MANDATORY OPTION	
((OPTION_C)) This is optional	Trie is mapped to directory
((OPTION_D)) This is optional	All of the above
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	Only B and C
((CORRECT_CHOICE)) Either A or B or C or D or E	E
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	What happens in case of overflow in extensible hashing,
((OPTION_A)) THIS IS MANDATORY OPTION	Buckets are splitted
((OPTION_B)) THIS IS ALSO	No of entries of Directory doesn’t change

MANDATORY OPTION	
((OPTION_C)) This is optional	Directory is doubled
((OPTION_D)) This is optional	Both A and C
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	D
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Which of below technique have less no of collision?
((OPTION_A)) THIS IS MANDATORY OPTION	Linear probing
((OPTION_B)) THIS IS ALSO	Quadratic probing

MANDATORY OPTION	
((OPTION_C)) This is optional	Chaining
((OPTION_D)) This is optional	Both A and B
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	B
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Pair below 1. Linear probing a. H(key)=(Hash(key)+i)mod max) 2. Quadratic probing b. H(key)=(Hash(key)+i^2)mod max) 3. Double hashing c. table is resized whose size is prime number 4. Rehashing d.H2(key)=M-(key mod M) 5. Chaining e. Few table indices may leave empty
((OPTION_A)) THIS IS MANDATORY OPTION	1-a,2-b,3-c,4-d,5-e

((OPTION_B)) THIS IS ALSO MANDATORY OPTION	1-b,2-a,3-c,4-d,5-e
((OPTION_C)) This is optional	1-a,2-b,3-d,4-c,5-e
((OPTION_D)) This is optional	1-c,2-b,3-a,4-e,5-d
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Which of below is true? 1. Skip list are better than link list in searching 2. Quadratic probing have less number of collision compared to linear probing 3. Resizing is cheap operation
((OPTION_A)) THIS IS MANDATORY OPTION	I only

((OPTION_B)) THIS IS ALSO MANDATORY OPTION	I and ii only
((OPTION_C)) This is optional	All of the above
((OPTION_D)) This is optional	Only i and iii
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	B
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Which of the below stores data in sorted order?
((OPTION_A)) THIS IS MANDATORY OPTION	Hash table

((OPTION_B)) THIS IS ALSO MANDATORY OPTION	Dictionary
((OPTION_C)) This is optional	Skip list
((OPTION_D)) This is optional	Both B and C
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	All of the above
((CORRECT_CHOICE)) Either A or B or C or D or E	D
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	Time complexity of perfect hash function in worst case is
((OPTION_A)) THIS IS MANDATORY OPTION	O(n)

((OPTION_B)) THIS IS ALSO MANDATORY OPTION	O(log n)
((OPTION_C)) This is optional	O(1)
((OPTION_D)) This is optional	O(n/2)
((OPTION_E)) This is optional. If optional keep empty so that system will skip this option	
((CORRECT_CHOICE)) Either A or B or C or D or E	C
((EXPLANATION)) This is also optional	

((MARKS)) QUESTION IS OF HOW MANY MARKS? (1 OR 2 OR 3 UPTO 10)	1
((QUESTION)) ENTER CONTENT. QTN CAN HAVE IMAGES ALSO	
((OPTION_A)) THIS IS MANDATORY OPTION	