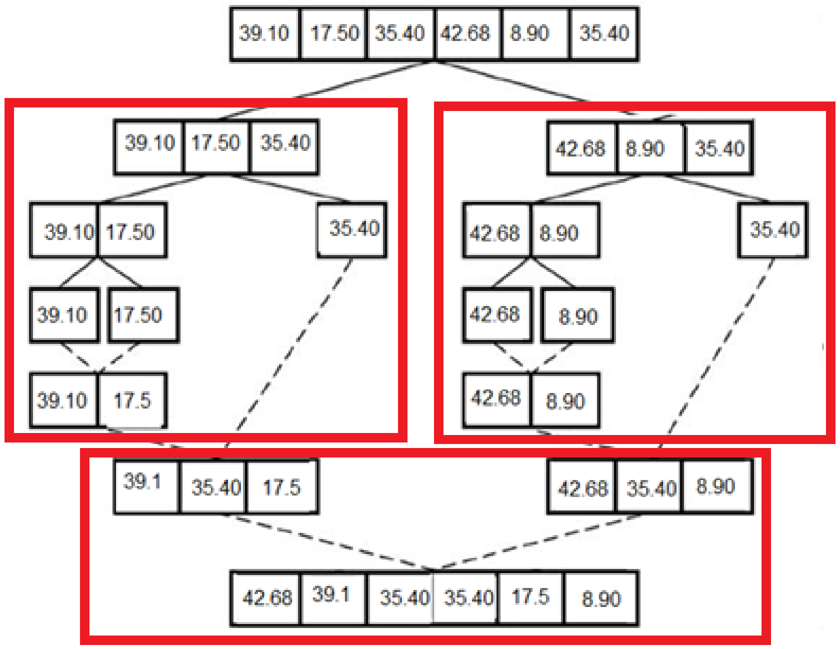
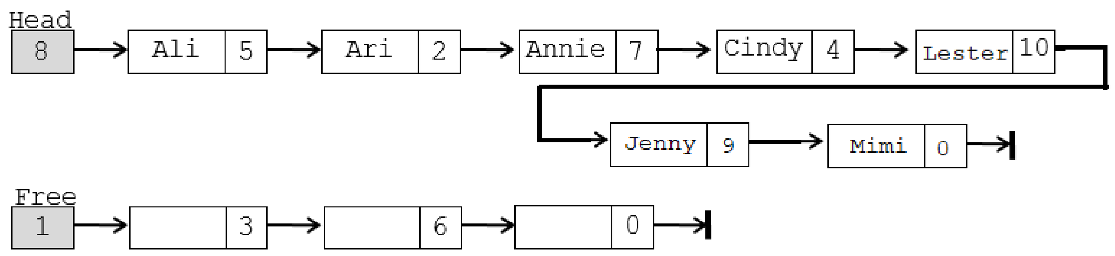
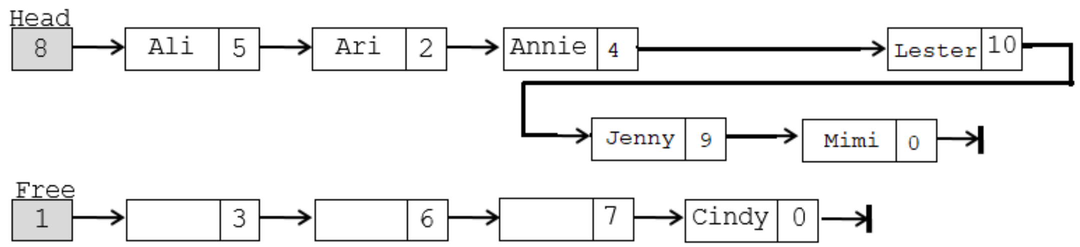
		2020 JC 2 H2 Computing EOY Solutions
1	**(a)**	Line 5.
		Index out of range which results in a runtime error at `X[i+1]`, when `i = 5`
	(b)	Line 5: `IF X[i] > X[i-1]`
		<u>**OR**</u>
		Line 4: `FOR i = 1 to N-1`
	(c)	

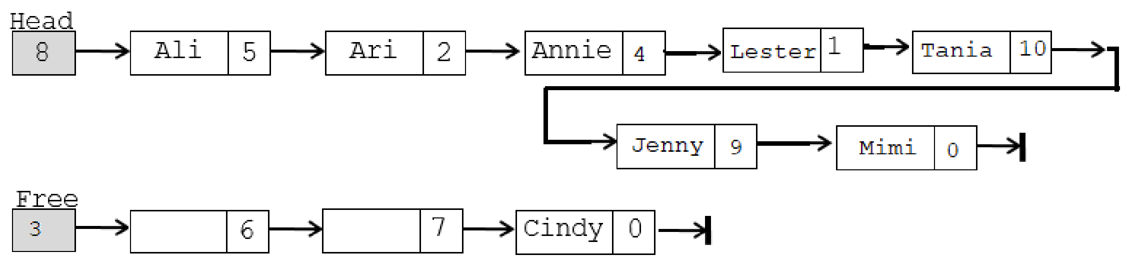
flag	i	X[1]	X[2]	X[3]	X[4]	X[5]
False	1	49.01	58.61	48.54	59.32	49.78
True	2	49.01	48.54	58.61	59.32	49.78
True	3	49.01	48.54	58.61	59.32	49.78
True	4	49.01	48.54	58.61	49.78	59.32
False	1	48.54	49.01	58.61	49.78	59.32
True	2	48.54	49.01	58.61	49.78	59.32
True	3	48.54	49.01	49.78	58.61	59.32
True	4	48.54	49.01	49.78	58.61	59.32
False	1	48.54	49.01	49.78	58.61	59.32
False	2	48.54	49.01	49.78	58.61	59.32
False	3	48.54	49.01	49.78	58.61	59.32
False	4	48.54	49.01	49.78	58.61	59.32

	(d)	Worst case scenario happens when the input race timings were pre-sorted in descending order of timings.
		$O(n^2)$
	(e)	10^2 or 10000 comparisons
2	**(a)**	A serial file is a file that stores information in <u>chronological order.</u>
	(b)	Backup file is a **duplicate of the current working file** intended to be **used as safety precaution/prevention against loss** of data due to unintended data corruption or data loss.
		Archive file is intended to **store data that not actively used but kept for historical references or auditing purposes.**
	(c)	x = 0, y = 5
	(d)	Total no of times `mergesort` is called = 11

		Total no of time `merge` is called = 5
	(e)	 39.10 17.50 35.40 42.68 8.90 35.40 39.1 17.5
	(f)	Regardless of the any possible cases `mergesort` will consistently divide the array by half recursively until the base case of 1 item remains, and then take linear time to recursively `merge` the two halves.
3	**(a)**	A recursive algorithm is a method of programming where the algorithm to a function or procedure is: • able to call itself one or more times in its body, and • terminates when it reaches its base case.
	(b)	Mention of • performing sequence of instructions repeatedly using loops • repetition continues unit condition fulfilled.
	(c)	`FUNCTION SumOfCubes(n) RETURNS INTEGER`

		`        IF n = 1`
		`          THEN`
		`            RETURN n`
		`        ENDIF`
		`RETURN n*n*n + SumOfCubes(n-1)`
		`ENDFUNCTION`

	(d)	`n` getting too large may result in stack memory overflow which in turn result in a runtime error, eventually crashing the program.
4	**(a)**	A linked list is a linear collection of data element whose: • orders are not given by their physical placement in the memory, • makes use of pointers to connect one element to another, joining all the nodes to form a sequence
	(b)	`a=7,    b=10,    c=2,    d=4,    e=0,    f=9,    g=5`
	(ci)	Head =8 Free = 1  Head 8 Ali 5 Ari 2 Annie 7 Cindy 4 Lester 10 Jenny 9 Mimi 0 Free 1 3 6 0
	(cii)	Head = 8 Free = 1  Head 8 Ali 5 Ari 2 Annie 4 Lester 10 Jenny 9 Mimi 0 Free 1 3 6 7 Cindy 0
	(ciii)	Head = 8 Free = 3



(d) **Linked list is a <u>dynamic data structure:</u>** it can grow and shrink at runtime by allocating and deallocating memory. So there is <u>no need to give initial size of linked list</u>.
Ease of insertion/deletion: Inserting a new element in an array of elements is expensive; because room has to be created for the new elements and to create room existing elements have to shift. Deletion is also expensive with arrays until unless some special techniques are used. For example, to delete 1010 in id [], everything after 1010 has to be moved.
No memory wastage: As size of linked list can increase or decrease at run time so there is no memory wastage. In case of array there is lot of memory wastage, like if we declare an array of size 10 and store only 6 elements in it then space of 4 elements are wasted. There is no such problem in linked list as memory is allocated only when required.

Disadvantages:
Pointer requires **extra memory** for storage.
No direct random access

5 **(a)**

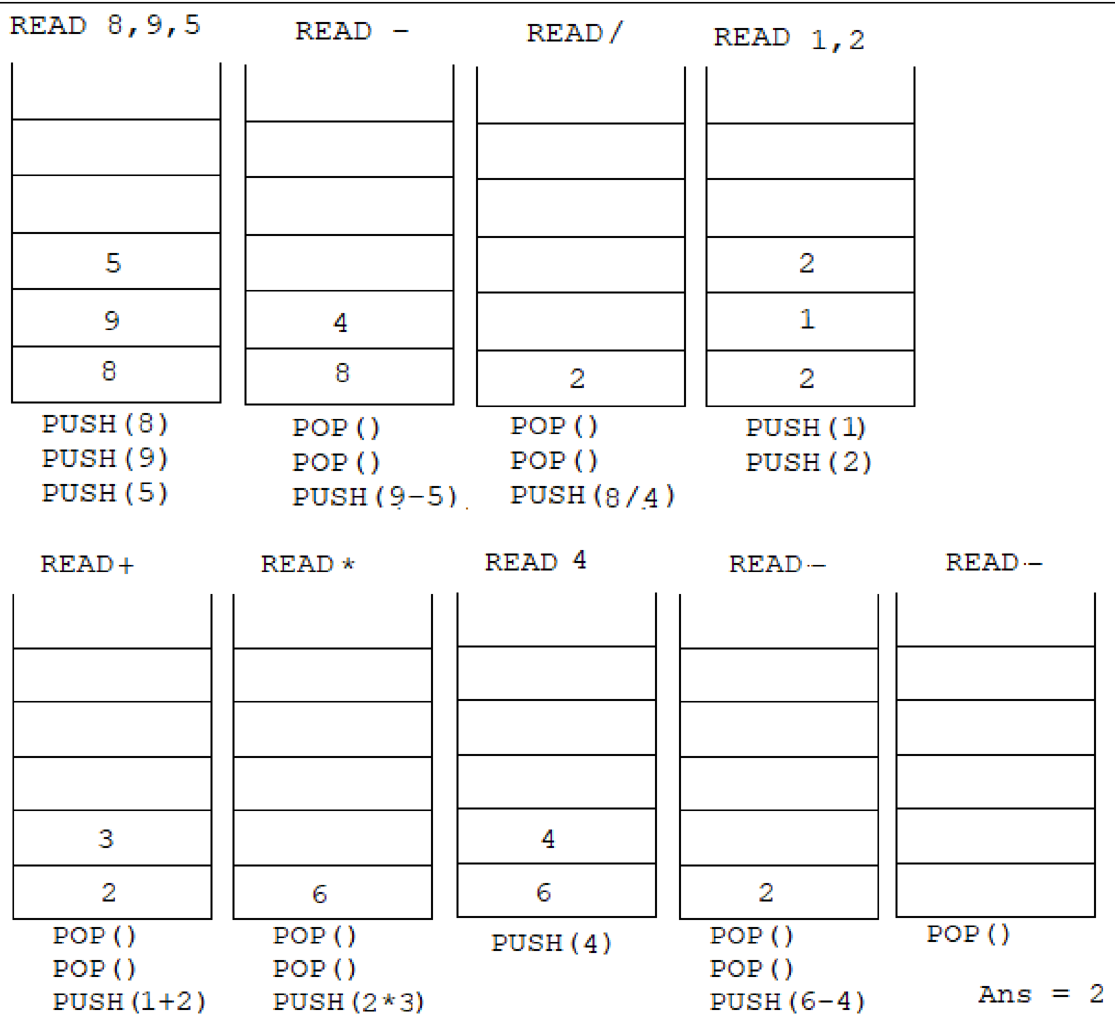
```
PROCEDURE PUSH(X):
    IF sp = 20 THEN
        RETURN NULL //stack is full
    ENDIF
    sp ← sp + 1
    S[sp] ← X
ENDPROCEDURE
```

```
FUNCTION POP():
    IF sp = 0 THEN
        RETURN NULL //stack is empty
    ENDIF
    temp ← S[sp]
    sp ← sp – 1
    RETURN temp
ENDFUNCTION
```

```
FUNCTION PEEK():
    IF sp = 0 THEN
        RETURN NULL //stack is empty
    ENDIF
    RETURN S[sp]
ENDFUNCTION
```

(b)

token	Description	STRING postfix	Stack, S
A	Appends to postfix	A	empty
/	Push to S	A	/
(	Push to S	A	/,(
B	Appends to postfix	AB	/,(
-	Push to S	AB	/,(,-
C	Appends to postfix	ABC	/,(,-
)	POP S and append to postfix Until '(' seen	ABC-	/
*	POP S and append to postfix Until S empty	ABC-/	*
D	Appends to postfix	ABC-/D	*
^	PUSH to S (since ^ is higher than *)	ABC-/D	*,^
E	Appends to postfix	ABC-/D	*,^
empty	POP S and append to postfix	ABC-/D^	*
empty	POP S and append to postfix	ABC-/D^*	empty

	(c)	 READ 8,9,5 5 9 8 PUSH(8) PUSH(9) PUSH(5) READ - 4 8 POP() POP() PUSH(9-5) READ / 2 POP() POP() PUSH(8/4) READ 1,2 2 1 2 PUSH(1) PUSH(2) READ + 3 2 POP() POP() PUSH(1+2) READ * 6 POP() POP() PUSH(2*3) READ 4 4 6 PUSH(4) READ - 2 POP() POP() PUSH(6-4) READ - POP() Ans = 2
6	**(ai)**	In-order: H, E, B, F, A, C, I, G, J
	(aii)	Pre-order: A, B, E, H, F, C, G, I, J
	(aiii)	Post-order: H, E, F, B, I, J, G, C, A.
	(bi)	*assume that the nodes of BST are all properly ordered properly. For every node in BST, starting with the root: 1. Compare current node with `search_key`. If they are equal, `RETURN TRUE`.

2. If value of `search_key` is less than value of current node, proceed to investigate nodes in the left subtree of current.
3. If value of `search_key` is more than value of current node, proceed to investigate nodes in the right subtree of current.
4. Repeat steps 1 to 3 until:
    a. value of `search_key` = value of current node, `RETURN TRUE`.
    b. current node has reached a leaf node, `RETURN FALSE`.

(bii) **Advantage of BST over linked list:**
Traversal of linked list may end up to be a linear sequential search with time complexity O(n). Traversing a BST on average takes O(log2n) time complexity, which is much more efficient as compared to linear search.

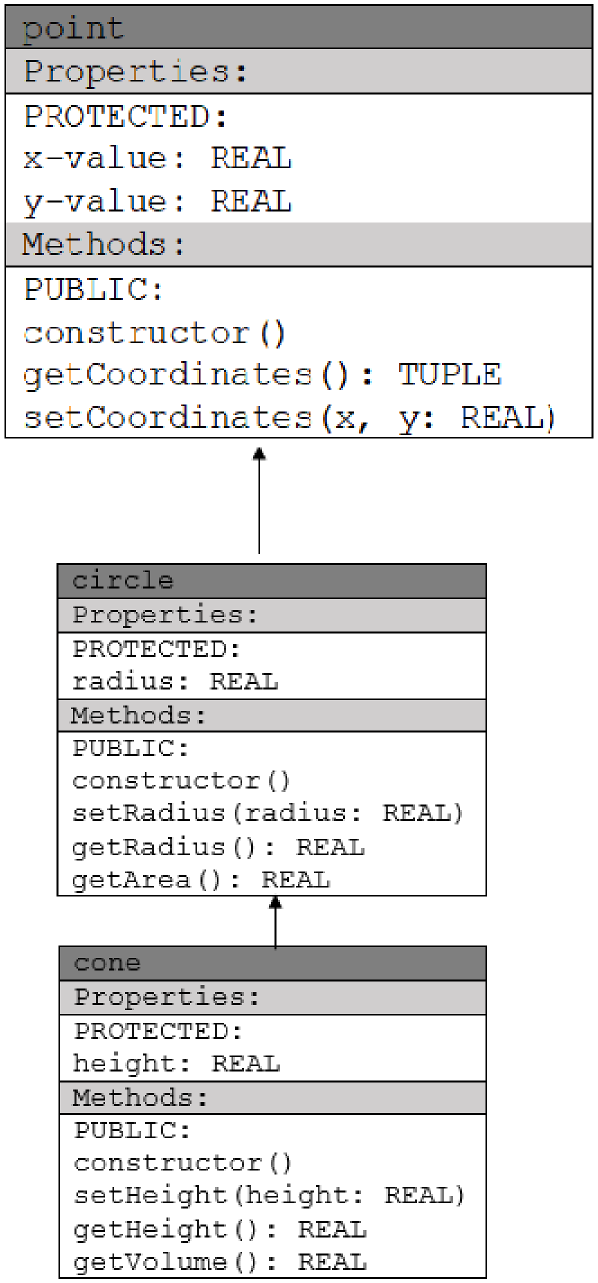
However, it would be undesirable, if the BST becomes totally skew left or right as a result of node insertions. A BST that is skew left or right totally will become like a linked list. ie. traversing through the BST will end up becoming a linear search.

7

Conditions												
Not more than 6 demerit points before deduction (ie. <=6)	Y	N	N	N	N	N	N	N	N	N	N	N
7 demerit points before deduction.		Y	Y	Y	Y							
8 demerit points before deduction.						Y	Y	Y	Y			
9 demerit points before deduction.										Y	Y	Y
NCD for past 3 years (carries deduction of 2 points)	-	N	Y	-	-	Y	N	N	N	Y	Y	N
Current employee of XYZ (carries deduction of 1 point)	-	N	-	Y	-	-	Y	Y	-	Y	-	-
CoM (carries deduction of 1 point)	-	N	-	-	Y	-	Y	-	Y	-	Y	-
Actions (<=6 points to be eligible)												
Eligible to purchase	**Y**		**Y**	**Y**	**Y**	**Y**	**Y**			**Y**	**Y**	
Not eligible to purchase		**Y**						**Y**	**Y**			**Y**

8 **(a)**
- Data independence: information is stored in such a way that any changes to the structure of the database will not affect the programs that access the data.

		• No redundant data as normalisation would have removed duplicates, thus saving storage space and access time. • Data consistency and integrity as a result of normalisation (duplicate data removed). • Data security and privacy: Supports use authentication, and also provide different levels of access rights to users to prevent data leaks, theft, or misuse. • Ease of access: Manages the information in • Data backup and recovery: backups data so that data can be fully restored when failure occurs.
	(b)	• Employee: DOB, contact number, address, email, etc • Claims: Date of claim, date of receipt, reasons for purchase, receipt number, etc. Any other points related to the context of this scenario.
9	**(a)**	$ABCD_{16}$ = $1010\ 1011\ 1100\ 1101_2$ = $2^0+2^2+2^3+2^6+2^7+2^8+2^9+2^{11}+2^{13}+2^{15}$ = 1+4+8+64+128+256+512+2048+8192+32768 = 43981_{10}
	(b)	4 bytes integer variable → 32 bits representation Total number of distinct representations = 2^{32} Since the variable is used to represent a discrete variable, [0, 2^{32}-1] Maximum positive integer = 2^{32}-1 = 4,294,967,295.

10	**(a)**	

point
Properties:
PROTECTED:
x-value: REAL
y-value: REAL
Methods:
PUBLIC:
constructor()
getCoordinates(): TUPLE
setCoordinates(x, y: REAL)
circle
Properties:
PROTECTED:
radius: REAL
Methods:
PUBLIC:
constructor()
setRadius(radius: REAL)
getRadius(): REAL
getArea(): REAL
cone
Properties:
PROTECTED:
height: REAL
Methods:
PUBLIC:
constructor()
setHeight(height: REAL)
getHeight(): REAL
getVolume(): REAL

	(bi)	private and protected members of a class • Private members of a class cannot be directly accessed by the member functions of the derived class. • Protected members of the base class can only be directly accessed by the member function of the derived class.
	(bii)	an object and a class. • An object is an instance of a class. • A class is a template for objects that have common attributes and methods.
11	**(a)**	Local area network: • Collection of devices in a small geographical are connected to one another. • The devices can be linked using twisted cables or Wi-Fi.
	(b)	Addition and removal of any node in a network is difficult and can cause issue in network activity in a LAN that is physically ring configured. Network failure diagnosis is complex and time consuming in a LAN that is physically ring configured as failure can possibly be due to cables or node. Ring network LAN with star set up physically streamlines diagnosis of network failure to a single point of error at the MAU.
	(c)	• The MSAU/MAU **physically** connects nodes in a **star topology** while retaining the network's **logical** ring structure. • Data transmission between nodes will follow a ring fashion. • The MAU/MSAU does not forward data from the sender node to its specified receiver node directly, but rather forwards data packet from one node to the next node repeatedly in a circular fixed sequence until it reaches at its proper destination.

	(d)	• Computer can only send data to printer when it has possession of the empty token. • When the computer is in possession of the empty token, it will change one bit in token to make SOF (start-of-frame) for data frame, and then append the data it wishes to send to the printer, printer’s address, its address, etc, into the frame. • The token will then be forwarded from one node to its next node in a fixed sequence until the intended recipient recognises this frame has its own address, copies the message, check for errors and **changes four bits in the last byte of the frame to indicate <u>address recognised</u> and <u>frame copied</u>**. • The full packet continues to circulate around the ring unit it returns to the sender computer, where it will examine the address recognised bits, if they are set, it knows the frame was in, discards the used data frame and release an emptu token back to the ring