Q.1) Max Marks: 1

Non-left recursive grammar for the following context free grammar is
S→ AA | 0
A→ SS | 1

A S→ AA| 0
A→ 0S | 1
A' → AS | ASA' | ε

B S→ AA| 0
A→ 1A' |0SA'
A' → AS | ASA'

C S→ AA| 0
A→ 0SA' | 1A' | 0S | 1
A' → ASA' |AS

Correct Option

Solution: (C)

Explanation:
Given grammar is
S→ AA | 0
A→ SS | 1
Contains indirect left recursion
S→ AA | 0
A→ AAS |0S |1
Eliminate the left recursion
S→AA | 0
A→ 0SA' | 1A'
A' → ASA' | ε
Eliminate ε- production
S→AA | 0
A→ 0SA' | 1A' | 0S | 1
A' → ASA' | AS

D S→ AA| 0
A→ 1A' |0SA' | 1 | 0
A' → ASA' | AS

Q.2) Max Marks: 1

Basic pipelining Use the following code fragment:
LW R1, 0(R4) ;R1 ← address (0+R4)
ADDI R2, R1, #8 ;R2 ← R1+8
MULT R3, R1, R1 ;R3 ← R1*R1
SW R3, 4(R2) ;address(4+R2) ← R3
Number of RAW (read-after-write) pipeline hazards in the code, regardless of whether they cause any stalls are_____

A 2

B 3

C 4

Correct Option

Solution: (C)
Explanation:
I1→I2 for R1
I1→I3 for R1
I2→I4 for R2
I3→I4 for R3

D None of these

Q.3) Max Marks: 1

A and B entered into a partnership with capitals in the ratio $4:5$. After 3 months, A withdrew $\frac{1}{4}$ of his capital and B withdrew $\frac{1}{5}$ of his capital. The gain at the end of 10 months was ₹ 760. A's share in this profit is:

A ₹ 330

Correct Option

Solution: (A)

Solution: Let the total capital invested be x.

Profit Sharing Ratio of A and B after 10 months will be:

$[(4x * 3) + (4x - \frac{1}{4} * 4x) * 7] : [(5x * 3) + (5x - \frac{1}{5} * 5x) * 7]$

$\Rightarrow [12x + (3x * 7)] : [15x + (4x * 7)]$

$\Rightarrow [12x + 21x] : [15x + 28x]$

$\Rightarrow 33x : 43x$

$\Rightarrow 33 : 43$

Therefore, A's share is $760 * \frac{33}{76} = ₹ 330$

B ₹ 380

C ₹ 360

D ₹ 430

Q.4) Max Marks: 1

A doubly linked-list of 6 nodes is given and one pointer ptr is pointing at third node then find the correct sequence for inserting new element after that.

A **new → next = ptr → next**
Ptr → next = new
New → prev = ptr
New → next → prev =new

Correct Option

Solution: (A)

Solution:

We are inserting new node after ptr. As we know that doubly linked list has three fields: data, prev node address and next node addresses

Initially, doubly linked list of 6-nodes are given as and let address is 1000, 2000, 3000, 4000, 5000, 6000. Head previous is null and tail next is null. Let new node address is 7000.

new → next = ptr → next // new → next = 4000 : in this line, we are assigning next address for new node as next of third node.

Ptr → next = new // Ptr → next = 7000 : by this line, we are making next address of ptr as new node because we are instructed to add new node after ptr.

New → prev = ptr // New → prev = 3000 : in this, we are making previous address of new node as address of ptr and new node is added just after the ptr so ptr is the previous node of new node.

New → next → prev =new // New → next → prev = 7000 : by this, we are making next of new that is node S with address 4000, we are putting address of its previous as new node as new is node just before this node.

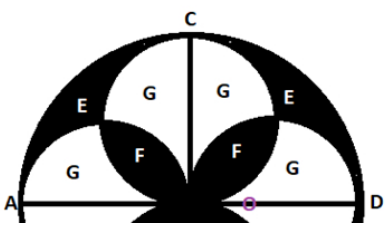
B **Ptr → next = new**
New → prev = ptr
New → next → next =new
new → prev = ptr → next

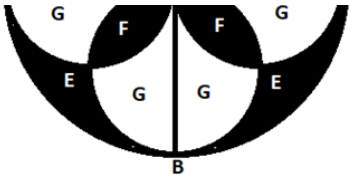
C **New → next = ptr**
New → next → prev =new
new → next = ptr → next
Ptr → prev = new

D **None of these**

Q.5) Max Marks: 1

ABDC is a circle and circles are drawn with AO, CO, DO and OB as diameters. Areas E and F are shaded. E/F is equal to





A 1 Correct Option

Solution: (A)

Solution: $AO = CO = DO = OB$ = radius of bigger circle = r (let)
Then area of $(G+F) = \frac{\pi r^2}{2}$
Area of $2(G+F) = \pi r^2$. Also area of $2G+F+E = \pi r^2$
i.e., $2G+F+F = 2G+F+E \Rightarrow F = E$
Hence, $E/F = 1$.

B 1/2

C $\pi/2$

D $\pi/4$

Q.6) Max Marks: 1

A binary search tree is given and we find its preorder traversal as follows:
Pre-order: 8, 3, 2, 1, 5, 6, 18, 12, 9, 29, 33
Then find out its post-order traversal and get the sum of elements of 2nd level of the constructed binary search tree(root is at level 0)

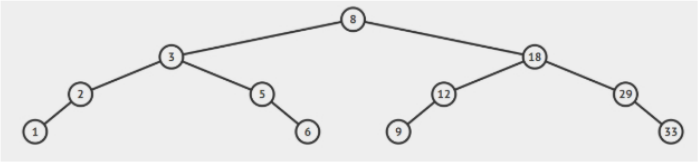
A 21

B 28

C 48 Correct Option

Solution: (C)

Solution: Ans is (C)48
Here, as we know that in binary search tree elements are in ascending order in its in-order traversal. So in-order will be 1, 2, 3, 4, 5, 6, 8, 9, 12, 18, 29, 33 and pre-order is given as **8, 3, 2, 1, 5, 6, 18, 12, 9, 29, 33.** So with the help in-order and preorder we can construct the tree which is as follows:



So if we count root at level 0 then at level 2 we get elements 2, 5, 12, 29 and sum of these four elements is 48.

So, the correct answer is 48.

D 49

Q.7) Max Marks: 1

Suppose a computer using a fully associative cache has 2^{16} bytes of memory and a cache of 64 blocks, where each block contains 32 bytes. To which cache set will the memory address 0xF8C9 map? (Assume that the set number starts from 1).

A 1st Correct Option

Solution: (A)

Explanation:
(2^{16} bytes of memory) / (32 bytes per block) = 2048 blocks
Because a fully associative cache has only one set, the set index will not require any bits. Because the associativity of the cache says that a memory location can map to any of the ways (blocks) in the cache, it is not meaningful to have a block offset. 32 bytes per block requires 5 bits of byte offset. There are 16 bits in the address all together, so the tag is 11 bits.
There is only 1 set, and that is the set that 0xF8C9 will map to.

B 2nd

C 4th

D 6th

Q.8) Max Marks: 1

The distance between the two towns is x km. A car travelling between the two towns covers the first k km at an average speed of y km/h and the remaining distance at z km/h. The time taken for the journey is

A $\frac{k}{y} + \frac{(x-k)}{z}$ Correct Option

Solution: (A)

Solution: The total time = time in the first part of the journey + time for the second part of the journey = $\frac{k}{y} + \frac{(x-k)}{z}$

B $ky + \frac{(k-x)}{z}$

C $\frac{k}{y} + \frac{(k-x)}{z}$

D $ky + z(x-k)$

Q.9) Max Marks: 1

A computer has 32-bit instructions and 12-bit addresses. If there are 250 two-address instructions, how many one-address instructions can be formulated?

A 2^{20}

B $6*2^{12}$ Correct Option

Solution: (B)

Explanation:

Address length is 32 bits

Possible two address instructions

Opcode	Add1	Add2
8	12	12

2^8 = 256 combinations.

Given that 250 two-address instructions

256 – 250 = 6 combinations can be used for one address.

Maximum number of one address instructions possible = $6*2^{12}$

C $8*2^{12}$

D **None of these**

Q.10) Max Marks: 1

Consider the following LL(1) grammar, which has the set of terminals T = {a, b, ep, +, *, (,)}. This grammar generates regular expressions over {a, b}, with + meaning the RegExp OR operator, and ep meaning the symbol.

E → TE'

E' → +E | ε

T → FT'

T' → T | ε

F → PF'

F' → *F' | ε

P → (E) | a | b | ep

Then Follow(P) =

A Follow(P) = {*, (, a, b, ep}

B Follow(P) = {(, a, b, ep, +,), $}

C Follow(P) = {(, a, b, ep, +,), *, $} Correct Option

Solution: (C)

Explanation:

First(E) = {(, a, b, ep} Follow(E) = {), $}

First(E') = {+, ε} Follow(E') = {), $}

First(T) = {(, a, b, ep} Follow(T') = {+,), $}

First(T') = {(, a, b, ep, ε} Follow(T') = {+,), $}

First(F) = {(, a, b, ep} Follow(F) = {(, a, b, ep, +,), $}

First(F') = {*, ε} Follow(F') = {(, a, b, ep, +,), $}

First(P) = {(, a, b, ep} Follow(P) = {(, a, b, ep, +,), *, $}

Follow(P) = First(F') ⇒ {*, ε} ⇒ {*} ∪ Follow{F} ={*, (, a, b, ep, +,), $}

D Follow(P) = {(, a, b, ep, +, *, $}

Q.11) Max Marks: 1

Number of tokens in the following C statement are_____

/*C-Comment*/ printf("GATE 2020%d", **a++);

Correct Answer

Solution: (10)

Answer: 10

Explanation:

Lexical analyser will ignore the comments

Printf	-1
(	-2
"GATE 2020%d"	-3
,	-4
*	-5
*	-6
a	-7
++	-8
)	-9
;	-10

Q.12) Max Marks: 1

A stack is implemented efficiently such that push and pop operations are performed in O(1) time. Then choose which of the following data structure is used.

Please note: In every list head is the pointer, which is pointing to the first node of list.

A **Singly linked-list**

B **Doubly linked-list**

C **Circular singly linked-list**

D **Circular doubly linked list** Correct Option

Solution: (D)

Solution: Circular doubly linked list.

As we know that stack follows LIFO means insertion and deletion is done at one end only. So if we do push and pop by using these data structure then

Singly linked-list: Here, we can not pop the last element in O(1) time as head pointer is pointing to the first element.

Doubly linked-list: For this from first element we can not get last in O(1) time as it can go to the next but its prev is null.

Circular singly linked-list: For this, we can not get last element in O(1) time as list is singly circular so from first element we can not go to the last element.

Circular doubly linked list: By using this, we can go to the last element as well as second so this is the data structure by using we can do push and pop in O(1) time we use circular doubly linked list then in this first node has address of second node and last node. So we can do by using circular doubly linked list in O(1) time.

Q.13) Max Marks: 1

A bonus of ₹ 1, 00, 000 was divided among 500 workers of a factory. Each male worker gets ₹ 100 and each female worker gets ₹ 500. Find the number of female workers in the factory.

A **250**

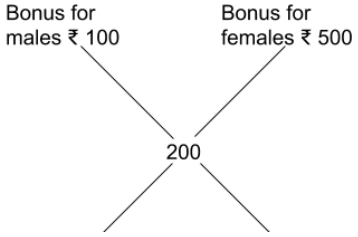
B **375**

C **290**

D **125** Correct Option

Solution: (D)

Solution: Average bonus per employee = 100000/500 = ₹ 200



Males : Females

500-200=300 200-100=100

3 : 1

Total Female workers = $500 * (1/4) = 125$.

Q.14) Max Marks: 1

The languages generated by the given context-free grammar.
S→ aSa | X
X→ aXa | bXb | ε

A $L = \{a^n a^m \mid m,n \geq 0,\ m=n\} \cup \{ww^R \mid w \in \{a,b\}^*\}$

B $L = \{a^n wwa^n \mid n \geq 0;\ w \in \{a,b\}^*\}$

C $L = \{a^n ww^R a^n \mid n \geq 0;\ w \in \{a,b\}^*\}$ Correct Option

Solution: (C)

Explanation:
S→ aSa will generate equal number of a's ⇒ $a^n X a^n$ (as S→ X also)
X→ aXa | bXb | ε ⇒ X⇒ $\{ww^R \mid w \in \{a,b\}^*\}$
Substitute X
The language generated by the grammar is
$L = \{a^n ww^R a^n \mid n \geq 0;\ w \in \{a,b\}^*\}$

D None of these

Q.15) Max Marks: 1

For a direct-mapped cache design with 32-bit address, the following bits of the address are used to access the cache

Tag	Index	offset
31-10	9-4	3-0

The ratio between total bits required for such a cache implementation over the data storage bits is____ (Including one valid bit)

A 1.14

B 1.18 Correct Option

Solution: (B)

Answer: 1.18
Explanation:
The data stored in a cache line needs 4 offset bits, so there are 16 bytes, or 4 words stored in a cache line
6 bits are used for the index, so there are 2^6 = 64 entries
A cache line requires 22 tag bits 1 valid bit, and 16 bytes for data.
The total storage required for the cache = 2^6(22 + 1 + 16*8) = 9664.
The amount of actual data stored in the cache is 2^6(16*8) = 8192.
Therefore the ratio is 9664/8192 = 1.18

C 1.20

D None of these

Q.16) Max Marks: 2

Assume that a computer with 34-bit memory address and 16KB 8-way set associative cache. If the cache block size is 32-bytes. Then the size of the tag memory(in bytes) including one valid bit.

Correct Answer

Solution: (1536)

Answer: 1536
Explanation:
Number of blocks in the cache = 16KB/32B = 512
Number of sets in the cache (8-way set associative) = 512/8 = 64
Number of bits in the index = 6
Address =34 bits

TAG	Set Index	Offset
23	6	5

Each cache line contains one valid bit.
Total TAG memory including valid bit is = (23+1)*Number of cache lines = 24*512
=12288 bits ⇒ 1536B

Q.17) Max Marks: 2

A set of 10 pipes (set X) can fill 70% of a tank in 7minutes. Another set of 5 pipes (set Y) fills 3/8 of the tank in 3minutes. The third set of 8 pipes (set Z) can empty 5/10of the tank in 10minutes. If one pipe is added for set X and set Y and set Z's capacity is increased by 20% on its original value and all the taps are opened at 2:58 p.m., then at what time does the tank get filled? (if it is initially empty.)

A **3:05 p.m**

B **3:04 p.m**

C **3:10 p.m**

D **3:03 p.m** Correct Option

Solution: (D)
Solution: Set X can fill 10% in a minute. Hence, every Pipe of set X can do 1% work per minute. Set Y has a filing capacity of 12.5% per minute (or 2.5% per minute for each tap in set Y). Set Z has a capacity of emptying the tank at the rate of 5% per minute and each tap of set Z can empty at the rate of 0.625% per minute.
The rate per minute with the given changes (in percentage terms) would be:
Set X = 11%, Set Y = 15% and Set Z = -6%
Hence, the net rate of change = 11+15-6 = 20% per minute and it would take 5minutes for the tank to fill. If all pipes are opened at 2:58 p.m., the tank would get filled at 3:03 p.m.

Q.18) Max Marks: 2

If a full n-ary tree is given and depth of this tree is mentioned as 'd' and no. of leaf nodes of the tree are represented as n^d then find the total number of nodes in the given tree, when n= 5 and d=4.

A **260**

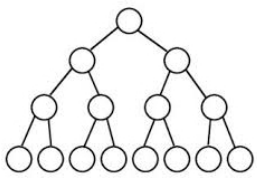
B **310**

C **626**

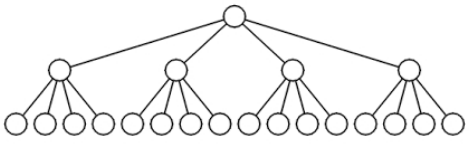
D **781** Correct Option

Solution: (D)
Solution: Ans is 781
For full n-ary tree, we can build our own logic by drawing tree and fetching total number of nodes or we need to use the formula which is as:
Total nodes = [(n * L) - 1] / (n-1) , where L is n^d

Full Binary Tree



In the given, full binary tree we can see that d=3 and n=2 so leaf nodes n^d = 2^3 = 8.
So total nodes are 15, verify given formula: [(n * L) - 1] / (n-1) = 2*8 -1 / 1 = 15



For given full 4-ary tree, we can see that n=4 and d= 2 so leave nodes = n^d = 4^2 = 16
So total nodes are 21, verify given formula: [(n * L) - 1] / (n-1) = [4*16 -1]/3 = 63/3 = 21

After verifying formula, we can apply this as:
L = n^d = 5^4 = 625
Total nodes = 5*625 - 1/ (5-1) = 3125-1 / 4 = 3124/4 = 781.
So, the correct answer is 781.

Q.19)

Max Marks: 2

Consider the following elements which are inserted into empty AVL.

34, 19, 24, 26, 39, 44, 49, 30, 37

Find the total number of points, which we are getting while constructing AVL tree. Assume there will be RR, LL, LR and RL rotations. And 2- points will be earned for every RR and LL rotation while 1- point will be earned for every RL and LL rotation.

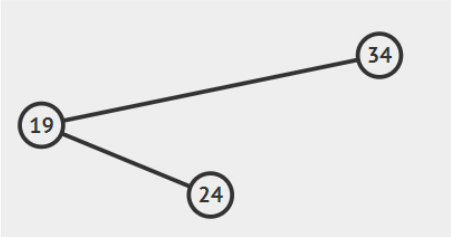
A 5 Correct Option

Solution: (A)

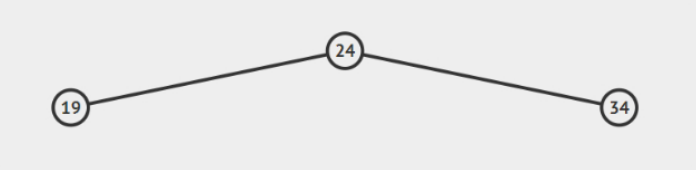
Solution: Ans is 5

AVL tree is the height balanced tree and every node in AVL tree can have balancing factor 0,+1 and -1. Here, we are saying, there will be four types of rotations while we are constructing the AVL tree. And we get the point for rotation of nodes.

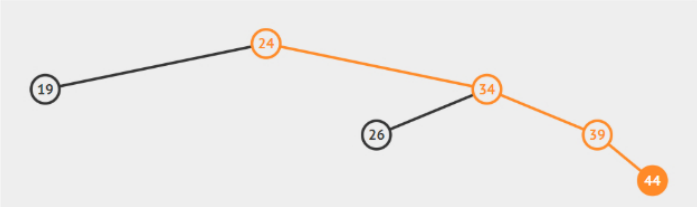
While we insert nodes 34, 19, 24. Initially tree will look like



1 rotation will be done. = 1 point

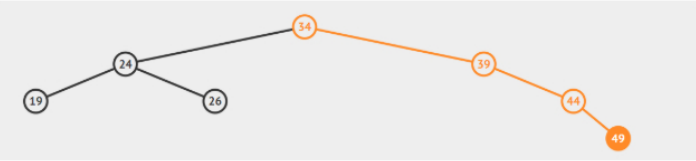


While, we insert 26, 39, 44 in the previous tree then after inserting 44, we can see that tree is not balanced. As balance factor for 39 is -1, balance factor for 34 is -1 and balance factor for 24 is -2.

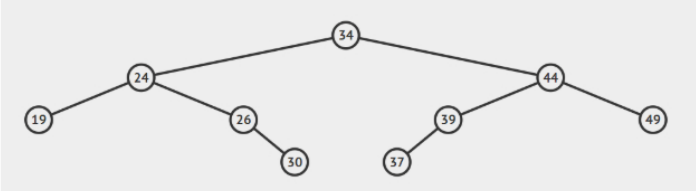


After balancing, tree will be like this

After inserting 49, tree is not balanced.



After inserting 30, 37 there will be no issue in balancing factor so final tree will look like



We are rotating tree while inserting elements 24, 44, 49. Because in this there will be 2 RR / LL rotations so 2*2 points will be earned here.
1 RL / LR rotation so 1 point will be earned here.

So total points will be 4+1 = 5.

B 4

C 3

D 6

Q.20) Max Marks: 2

There are two stacks S and T are given in which stack S is containing elements 1,2,3,4,5 (top) while stack T is empty.

```
Call ( )
{
while(!S isEmpty())
T.push(S.pop());
while(!T isEmpty())
Q.enqueue(T.pop());
while(! Q isEmpty());
S.push(Q. dequeue());
}
```

After going through the code, what will be the order of elements in the Stack S.

A (top)1,2,3,4,5

B (top)2,1,4,3,5

C (top)5,4,3,2,1 Correct Option

Solution: (C)

Solution: Ans is (top) 5,4,3,2,1
We know that Stack works on LIFO(Last in first out) and queue works on FIFO(first in first out). Here, simply we are deleting elements from stack S and inserting them into stack T and then inserting these elements into Q queue and then removing elements from queue and inserting back to the Stack S.

Initially S: 1, 2, 3, 4,5(top)
Call ( ) // starting function body
{
while(IS isEmpty()) // condition is true till S is not empty.
T.push(S.pop()); // This line will be executed until the while loop condition is failing.
// pop from S and push into T. T: 5, 4, 3, 2, 1(top)

while(IT isEmpty()) // condition will be true until T is empty.
Q.enqueue(T.pop());//This line will be executed until the while loop condition is failing.
pop from T and enqueue into Q. Q: (head) 1, 2, 3, 4, 5

while(I Q isEmpty());// condition will be true until Q is empty.
S.push(Q. dequeue()); // This line will be executed until the while loop condition is failing. Dequeue from Q and push back to S. S: 1, 2, 3, 4, 5(top)

So the correct answer is (top) 5,4,3,2,1.

D (top)4,5,1,2,3

Q.21) Max Marks: 2

Suppose processor X executes instructions in the following 4 stages (no pipeline), where each stage could run this fast:

IF&ID	10ns
EX	5ns
MEM	20ns
WB	5ns

Now assume that 20% of the instructions are load/save instructions. The remaining instructions do not need to execute the MEM stage. The performance of a pipelined processor compared to the non pipelined implementation of processor X

Correct Answer

Solution: (1.2)
Answer: 1.2
Explanation:
Non-pipelined = 10+5+20+5 = 40ns/instruction
Pipelined = 20ns/instruction (assuming no stalls), stages take 20ns each
Given that 20% of the instructions are load/save instructions and remaining 80% instructions do not need to execute the MEM stage
Non-pipelined = .80*20ns + .20*40ns = 16+8 = 24ns/instruction (avg)

Pipelined = 20ns/instruction, same as before
So performance is improved by = 24/20 = 1.2

Q.22) Max Marks: 2

Gunpowder can be prepared by saltpetre and nitrous oxide. Price of saltpetre is thrice the price of nitrous oxide. Notorious gangster Kallu Bhai sells the gunpowder at ₹ 2160 per $10\,g$, thereby making a profit of 20%. If the ratio of saltpetre and nitrous oxide in the mixture be $2:3$, find the cost price of saltpetre.

A ₹ $210/gm$

B ₹ $300/gm$ Correct Option

Solution: (B)

Solution : Cost of making one gram of gunpowder $= ((2160 * 100)/120)/10 =$ ₹ 180. In one gram of gunpowder, we have $0.4\,g$ of saltpeter and $0.6\,g$ of nitrous oxide as per the given ratio of $2:3$. So now as per the given question we will check with the options given. So at the rate of ₹ $300/gm$ of Saltpetre, the cost of nitrous oxide will be $300/3 =$ ₹ $100/gm$. Hence, $0.4\,g$ of saltpetre will cost $300 * 0.4 =$ ₹ 120 and $0.6\,g$ of nitrous oxide will cost $100 * 0.6 =$ ₹ 60. Hence we get the total cost as $120 + 60 =$ ₹ 180.

C ₹ $120/gm$

D **None of these**

Q.23) Max Marks: 2

Consider the following translation scheme
$S \rightarrow ER$
$R \rightarrow * E\{print("+");\}R \mid \varepsilon$
$E \rightarrow F + E\ \{print(" * ");\} \mid F$
$F \rightarrow (S) \mid id\ \{print(id.\ value);\}$
Here id is a token that represents an integer and $id.\ value$ represents the corresponding integer value. What does this translation scheme print for the input: 6 + 7 * 9 + 2?

A 67+92+*

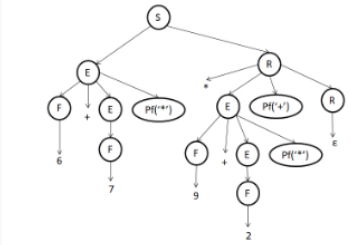
B 67+*92+

C 67*+92*

D 67*92*+ Correct Option

Solution: (D)

Explanation:



Q.24) Max Marks: 2

Consider the following grammar G.
S → aAb | Ac
A → a
Where S, A are non-terminal symbols, a, b and c are terminal symbols. Which of the following statement(s) is/are correct?
S1: LL(1) can parse all strings that are generated using grammar G.
S2: LR(1) can parse all strings that are generated using grammar

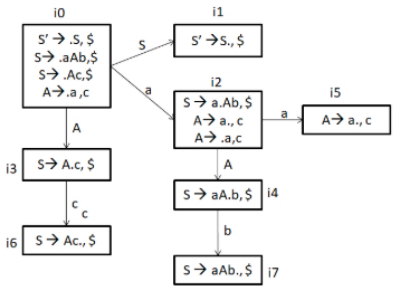
A **S1 only**

B **S2 only** Correct Option

Solution: (B)

Explanation:
S → aAb | Ac
A → a

First(S) = {a} ∩ {First(A)} = {a} ∩ {a} ={a} which is not equal to Φ
So LL(1) parse table contains multiple entries in the table.
It is not LL(1)



I2 contains shift and reduce operations.
First(A) ∩{c} = {a} ∩{c }= Φ No S/R conflict in LR(1). The given grammar is LR(1)

C **Both S1 and S2**

D **Neither S1 nor S2**

Q.25) Max Marks: 2

Media applications that play audio or video files are part of a class of workloads called "streaming" workloads; i.e., they bring in large amounts of data but do not reuse much of it. Consider a video streaming workload that accesses a 512 KB working set sequentially with the following address stream: 0, 2, 4, 6, 8, 10, 12, 14, 16,..... In general, cache access time is proportional to capacity. Assume that main memory accesses take 60 ns and that memory accesses are 32% of all instructions. Consider a 64 KB direct-mapped L1 cache with a 32-byte cache line. This L1 cache is attached to a processor and its hit time is 0.85 ns. What is the miss rate for the address stream above?

A **6**

B **6.25** Correct Option

Solution: (B)

Answer:6.25
Explanation:
The cache has 64 * 1024 / 32 = 2048 cache lines.
When byte 0 is accessed, we get a miss but we cache the entire 32-byte cache line, which implies that, all bytes from byte 0 till byte 31 (block 0) are cached.
Since the cache is direct-mapped, this block is cached at entry 0.
So far we got 1 miss and 15 hits(2,4,6,8,.....30).
The same trend continues for all 512*1024/32 = 16384 blocks of the 512 KB working set (except that starting from block 2048, we will have to start replacing blocks we had cached earlier).
The miss rate is 1/16 = 6.25%.
This miss rate is totally insensitive to the size of the cache and the size of the working set. The only factor that affects the hit rate is the cache line size. All the misses experienced by this workload are cold cache misses.

C **16**

D **32**

close