MADE EASY

India's Best Institute for IES, GATE & PSUs

# Important Questions for GATE 2022

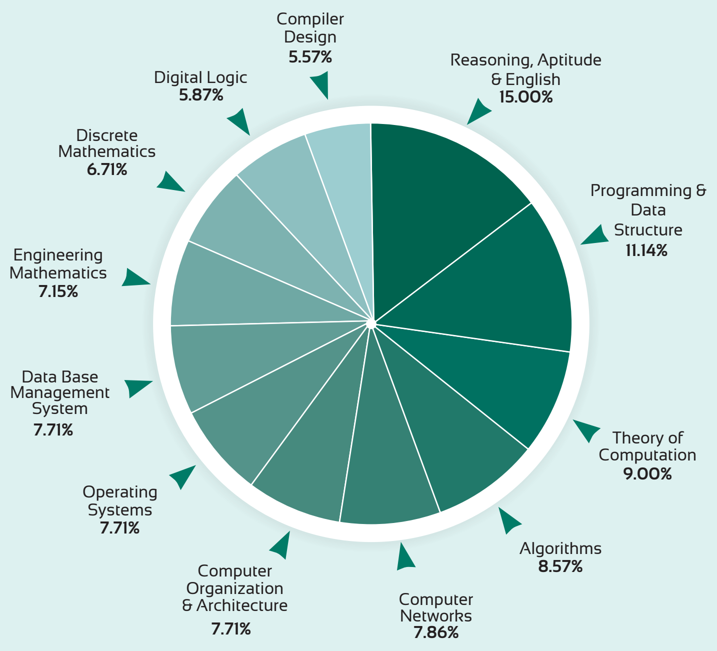
## COMPUTER SCIENCE & IT

## Day 6 of 8

Q.126 - Q.150 (Out of 200 Questions)

# Operating System + Compiler Design

## SUBJECT-WISE WEIGHTAGE ANALYSIS OF GATE SYLLABUS



| Subject | Average % (last 5 yrs) |
|---|---|
| Reasoning, Aptitude & English | 15.00% |
| Programming & Data Structure | 11.14% |
| Theory of Computation | 9.00% |
| Algorithms | 8.57% |
| Computer Networks | 7.86% |
| Operating Systems | 7.71% |
| Computer Organization & Architecture | 7.71% |
| Data Base Management System | 7.71% |
| Engineering Mathematics | 7.15% |
| Discrete Mathematics | 6.71% |
| Digital Logic | 5.87% |
| Compiler Design | 5.57% |
| **Total** | **100%** |

# Operating System + Compiler Design

**Q.126** Consider the following producer and consumer code:

```
# define N = 100
int mutex = 1;          // Binary semaphore variable
int empty = N;          // Counting semaphore variable
int full = 0;           // counting semaphore variable
```

```
Void producer ( )                        Void consumer ( )
{                                        {
    int item;                                int item;
    while (1)                                while (1)
    {                                        {
    item = producer-item ( );                down (mutex);
    down (empty);                            down (full);
    down (mutex);                            item = consume_item ( );
    insert_item (item);                      up (mutex);
    up (mutex);                              up (empty);
    up (full);                               }
    }                                    }
}
```

In the above code **mutex**, **empty** and **full** are semaphore shared variables and **item** is local to the both producer and consumer.

Insert_item (item) function will place "item" into buffer and consume_item function removes an item from the buffer. Which of the following holds true by the code?

(a) Satisfies mutual exclusion and no deadlock occurs
(b) Satisfies mutual exclusion but deadlock may occur
(c) Not satisfies mutual exclusion and no deadlock occurs
(d) Not satisfies mutual exclusion but deadlock may occur

**Q.127** Which of the following defines the spatial locality?

(a) If any memory location is accessed, the memory near to it will also be accessed.
(b) If any memory location is accessed, the same location will be accessed again soon.
(c) Both (a) and (b)
(d) Neither (a) nor (b)

**Q.128** Consider the following work load.

| Process | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|---|
| Arrival time | 0 | 3 | 1 | 2 | 2 |
| Burst time | 2 | 4 | 3 | 2 | 4 |
| Priority | 1 (highest) | 4 | 5 (lowest) | 2 | 3 |

Processes are scheduled with priority scheduler. Assume that priority scheduler is non-preemptive. What is the average turn around time with the priority scheduler?

**Q.129** Consider the following characteristics of a disk system.

- 10 GB disk rotates at 10,000 rpm
- Data transfer rate of $10^7$ bytes/sec
- Average seek time is 8 ms
- Block size is 32 KB

What is the average service time to retrieve a single disk block from a random location on the disk (in ms)?

**Q.130** Consider a system having demand paged system, where integers are stored in 4 bytes, page size is 256 bytes.

Consider the below program:

```
    int a[ ] [ ] = new int [200] [200];
        int i = 0;
        int j = 0;
 while (i + + < 200)
 {
          j = 0;
 while (j + + < 200)
     a[i] [j] = 0
 }
```

How many pages are needed to store the elements of an array?

(Assuming all elements of the array are stored in contiguous memory location?)

**Q.131** Consider a file system with the following information:

- Each block size 4096 bytes
- 32-bit disk block and file block pointers
- Each inode file has 13 direct, 4 singly-indirect, one double indirect, and one triply indirect pointers. Each inode maintains 52 bytes for other file information.
- Disk capacity is 16 TB.

How many inode files can fit into a single disk block when inode has only the above mentioned information?

**Q.132** Consider the following page reference stream.

a, b, c, a, b, d, a, d, b, c, b

How many page faults occurs using optimal (min) page replacement policy, if memory has 3 empty frames.

**Q.133** A process burst time can be using exponential averaging technique of shortest process next scheduling. Assume $E(t)$ is the estimation burst time of a process at time $t$, burst $(t)$ is the actual burst time of a process at time $t$, and $0 \le \alpha \le 1$.

What is the formula to predict the burst time of $t$?

(a) $E(t) = B(t) \cdot \alpha + (1 - \alpha) \cdot B(t - 1)$

(b) $E(t) = \alpha \cdot E(t - 1) + (1 - \alpha) \cdot E(t - 2)$

(c) $E(t) = \alpha \cdot B(t - 1) + (1 - \alpha) \cdot B(t - 2)$

(d) $E(t) = \alpha \cdot E(t - 1) + (1 - \alpha) \cdot B(t - 1)$

**Q.134** Consider the following code to solve the critical section problem for two processes $P_0$ and $P_1$. Initially flag [*i*] contain false for $i = 0$ and 1.

```
while (1)
{
    flag [i] = true;
    while (flag [j]);
    <critical section>
    flag [i] = false;
    <remainder section>
}
```

Assume $i$ refers to the current process $P_i$ and $j$ refers the other process $P_j$. If two processes executing above code concurrently then which of the following does not satisfy the above solution?

(a) Mutual exclusion and progress
(b) Mutual exclusion and bounded wait
(c) Progress
(d) None of these

**Q.135** Consider the following parameters of a disk system.

- Number of cylinders = $N_{\text{cylinder}}$
- Number of sectors/track = $N_{\text{sector}}$
- Number of surfaces = $N_{\text{surface}}$
- Number of tracks per surface = $N_{\text{track}}$
- Sector capacity = $C_{\text{sector}}$
- Cylinder capacity = $C_{\text{cylinder}}$

Which of the following is correct to compute the disk capacity of the system?

(a) Disk capacity = $N_{\text{surface}} \times N_{\text{track}} \times N_{\text{sector}} \times C_{\text{sector}}$
(b) Disk capacity = $N_{\text{cylinder}} \times C_{\text{cylinder}}$
(c) Disk capacity = $N_{\text{surface}} \times N_{\text{cylinder}} \times N_{\text{sector}} \times C_{\text{sector}}$
(d) All of the above

**Q.136** Consider the following system snapshot:

| Process | Currently Allocated | | | | Maximum | | | | Needed | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | U | V | W | X | U | V | W | X | U | V | W | X |
| A | 0 | 0 | 1 | 2 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 |
| B | 1 | 0 | 0 | 0 | 1 | 7 | 5 | 0 | 0 | 7 | 5 | 0 |
| C | 1 | 3 | 5 | 4 | 2 | 3 | 5 | 6 | 1 | 0 | 0 | 2 |
| D | 0 | 6 | 3 | 2 | 0 | 6 | 5 | 2 | 0 | 0 | 2 | 0 |
| E | 0 | 0 | 1 | 4 | 0 | 6 | 5 | 6 | 0 | 6 | 4 | 2 |

Given that currently available resources in the system as: (U, V, W, X) = (1, 5, 2, 2). How many total safe sequences are possible in the above system with 5 processes?

**Q.137** Consider the following segment:

```
int count = 0;
void tally( )
{
    for (int i = 1; i <=5; i++)
        count = count +1;
}
main( )
{
    parbegin
        tally( );
        tally( );
        tally( );
    parend
}
```

**Note:** Assume the count = count + 1; will execute in '3' different instructions.

**I.** Load $R_i$, m[count]

**II.** INC $R_i$

**III.** Store $m$[count], $R_i$

where $m$[count] refers to memory value of count variable.

Preemption can occur while executing the above instructions. Each of the tally function has a separate register R allocated to it in which the value of count is stored. After completion of main function, what will be the minimum and maximum values of count in the end of the program?

(a) min = 5, max = 15
(b) min = 2, max = 15
(c) min = 15, max = 15
(d) min = 3, max = 15

**Q.138** Consider a paged virtual memory system with 32-bit virtual addresses and 1 KB page size. Each page table entry requires 32 bits. It is desired to limit the page table size to one page. Assume multi level paging is used to implemented the above requirement then how many levels are required?

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

**Q.139** Consider the following augmented grammar with labels a and b:

$S' \to S$

**a:** $S \to (S)S$

**b:** $S \to \varepsilon$

LR (0) sets are given as following for the above grammar.

| (1) | (2) | (3) | (4) | (5) | (6) |
|---|---|---|---|---|---|
| $S' \to .S$<br>$S \to .(S)\ S$<br>$S \to .$ | $S' \to S.$ | $S \to (.S)\ S$<br>$S \to .(S)\ S$<br>$S \to .$ | $S \to (S.)\ S$ | $S \to (S).\ S$<br>$S \to .(S)\ S$<br>$S \to .$ | $S \to (S)\ S.$ |

If the following SLR (1) table is constructed as below then find the missing entries at $E_1$, $E_2$ and $E_3$ respectively.

| | ( | ) | \$ | S |
|---|---|---|---|---|
| 1 | $S_3$ | $r_b$ | $r_b$ | 2 |
| 2 | | | Accept | |
| 3 | $S_3$ | $E_1$ | $E_2$ | 4 |
| 4 | | $S_5$ | | |
| 5 | $S_3$ | $r_b$ | $r_b$ | $E_3$ |
| 6 | | $r_a$ | $r_a$ | |

(a) $r_a$, $r_b$, 5
(b) $r_a$, $r_a$, 6
(c) $r_b$, $r_a$, 5
(d) $r_b$, $r_b$, 6

**Q.140** Consider the following SDT:

$G \to A - T\ \{G.x = A.x - T.x\}$

$A \to Ea\ \{A.x = E.x \times 2\}$

$E \to Eb\ \{E_1.x = 1 + E_2.x\}$

$E \to \varepsilon\ \ \{E.x = -1\}$

$T \to Eb\ \{T.x = E.x - 2\}$

If above SDT uses L-attributed definition then what is the value of an attribute $x$ at root after evaluation for an input string "ba – bb".

**Q.141** Consider the following sets of items for some CFG produced while constructing LR (1) parser.

**Set 1(State):** $A \to a., \{b\}$
$B \to ba., \{b,c\}$

**Set 2 (State):** $A \to B.a, \{\$\}$
$B \to aB., \{b\}$

Each item has look ahead part which is computed using CLR (1) parser. Assume the grammar has follow sets as: **Follow (A) = {a, b} and Follow (B) = {b, c}**

Which of the following can be concluded from above two states of CLR (1) construction. [All other states donot conflict]

(a) Grammar is SLR (1)
(b) Grammar is CLR (1) but not SLR (1)
(c) Grammar is not CLR (1) due to SR conflict
(d) Grammar is not CLR (1) due to RR conflict

**Q.142** Consider the following assembly code:

MOV $R_1$, b ;    $R_1 \leftarrow b$
MOV $R_2$, c ;
MUL $R_1$, $R_2$;
MOV $t_1$, $R_1$;
MOV $R_1$, a ;
MOV $R_2$, $t_1$;
ADD $R_1$, $R_2$;
MOV $t_2$, $R_1$;
MOV $R_1$, $t_2$;
MOV d, $R_1$;

Find the correct expression which is equivalent to the above code.

(a) d = b + a * c
(b) d = a+b * c
(c) d = c + a * b
(d) d = a * b + a * b + c

**Q.143** Consider the following 3-address code:

$t_1 = t + e$
$t_2 = g + a$
$t_3 = t_1 * t_2$
$t_4 = t_2 + t_2$
$t_5 = t_4 + t_3$

There are five temporary variables in above code. Find minimum number of temporary variables that can be used in the equivalent optimized 3-address code of above code.

**Q.144** Let G be the following grammar:

$A \to AB \mid a$

$B \to *AC \mid Cb \mid \varepsilon$

$C \to +ABc \mid \varepsilon$

Find the total number of reductions using LR(1) parser for the string a∗a+ac using grammar G.

**Q.145** Consider the following grammar:

$S \to XX$

$X \to b$

$X \to aX$

Where S and X are non-terminals and a, b are terminals. Which of the following can be the viable prefixes?

(a) baab
(b) aab
(c) aaabab
(d) bbbaX

**Q.146** Consider the following C program:

**Program 1:**

```
int main ( )
{
    int a = 3;
    @ _ /* hello world */
    return 0;
}
```

**Program 2:**

```
int main ( )
{
    int x = 10;
    if (x > 20
    x = 20;
}
```

**Program 3:**

```
int main ( )
{
    int x = 20;
    /* hello world
    if (x > 20)
    x = 10;
}
```

**Program 4:**

```
int main ( )
{
    int a = 09;
    /* hello world */
}
```

Assume programs are compiled same as give above. How many programs results in lexical error?

(a) 1
(b) 2
(c) 3
(d) 4

**Q.147** Consider the given SDT having left recursion. Which of the following options is correct after elimination of left recursion from the SDT.

$X \rightarrow XY \{X.x = f(X.x, Y.y)\}$
$X \rightarrow Z \{X.x = g(Z.z)\}$

(a) $X \rightarrow Z \{X.x = g(Z.z)\}\ X'$
$X' \rightarrow Y \{X'.x = f(X.x, Y.y)\}\ X'$

(b) $X \rightarrow Z \{X.x = g(Z.z)\}\ R$
$R \rightarrow Y \{R.x = f(X.x, Y.y)\}\ R$
$R \rightarrow \in$

(c) $X \rightarrow Z \{X.x = g(Z.z)\}\ R$
$R \rightarrow \{R.x = f(X.x, Y.y)\}\ Y$
$R \rightarrow \in$

(d) None of the above

**Q.148** Consider the following grammar:

$S \rightarrow XaXb \mid YbYa$
$X \rightarrow \in$
$Y \rightarrow \in$

The above grammar is:

(a) LL(1) but not LR(1)
(b) LL(1) and LR(1)
(c) LR(0) but not LL(1)
(d) SLR(1) and LR(1)

**Q.149** Consider the intermediate code given below:

1. $i = 10$
2. $j = 1$
3. $a = i * j$
4. $b = i + j$
5. If $b \leq a$ goto 7
6. $a = a + 1$
7. $i = i - 1$ goto 3

X and Y are the number of nodes and edges in the control flow-graph constructed for the above code then the value of (X + Y) is ________.

## Multiple Select Question (MSQ)

**Q.150** Which of the following functions is/are performed by loader?

(a) Allocate memory for the programs and resolve symbolic references between objects decks.
(b) Address dependent locations, such as address constants, to correspond to the allocated space.
(c) Physically place the machine instructions and data into memory.
(d) Linking in the program.

■■■■

## Detailed Explanations

**126. (b)**

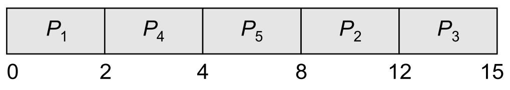
Given code correctly implements solve the mutual exclusion but producer and consumer may enter the deadlock.

Deadlock may occur; if consumer executes first down (mutex) and then down (turn), consumer goes to block mode. Now producer executes down (empty) and then down (mutex), producer goes to block mode. Deadlock may occur due to the above executions when buffer is initially empty.

**127. (a)**

Spatial locality refers option (a) and temporal locality refers option (b)

**128. (6.6)**



$$\begin{aligned}
\text{TAT}_1 &= 2 - 0 = 2\\
\text{TAT}_2 &= 12 - 3 = 9\\
\text{TAT}_3 &= 15 - 1 = 14\\
\text{TAT}_4 &= 4 - 2 = 2\\
\text{TAT}_5 &= 8 - 2 = 6
\end{aligned}$$

$$\text{Average TAT} = \frac{2+9+14+2+6}{5} = \frac{33}{5} = 6.6$$

**129. (14.28) [14.26 to 14.29]**

$$\begin{aligned}
T_{\text{seek}} &= 8 \text{ ms}\\
T_{\text{rotational}} &= \frac{1}{2} \text{ rotational latency}\\
&= \frac{1}{2}\left(\frac{60}{10000}\right) \text{sec} = 3 \text{ ms}\\
T_{\text{transfer}} &= \frac{32 \times 1024}{10^7} = 3.2768 \text{ ms} \approx 3.28 \text{ ms}\\
T_{\text{service}} &= 8 \text{ ms} + 3 \text{ ms} + 3.28 \text{ ms}\\
&= 14.28 \text{ ms}
\end{aligned}$$

**130. (625)**

$$\begin{aligned}
\text{Number of elements} &= 200 \times 200\\
\text{Memory needed} &= 200 \times 200 \times 4\text{B}\\
&= 160000\text{B}\\
\text{Number of pages needed} &= \frac{160000\text{B}}{256\text{B}} = 625
\end{aligned}$$

**131. (32)**

**Inode file has:**

| | | | |
|---|---|---|---|
| (i) | 13-direct pointers | $\Rightarrow$ | $13 \times 32$-bit = 52 bytes |
| (ii) | 4 singly indirect | $\Rightarrow$ | $4 \times 32$-bit = 16 bytes |
| (iii) | 1 Double indirect | $\Rightarrow$ | $1 \times 32$-bit = 4 bytes |
| (iv) | 1 triple indirect | $\Rightarrow$ | $1 \times 32$-bit = 4 bytes |

(v) 52 bytes for other information

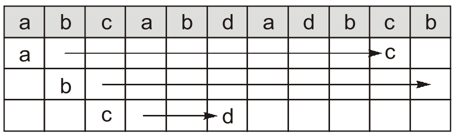
Each inode file has total 128 bytes

Given block size is 4096 bytes

$$\text{Number of inode files in each disk block} = \frac{4096}{128} = 32$$

**132. (5)**

Page reference:



5 page faults generated using optimal policy.

**133. (d)**

$E(t) = \alpha \cdot E(t-1) + (1-\alpha) \cdot B(t-1)$

where, $E(t-1)$ is the most recent expected burst time of a process.

$B(t-1)$ is the most recent actual burst time of a process.

**134. (c)**

- Mutual exclusion is satisfied by the given code.
- Progress does not satisfy because both $P_0$ and $P_1$ may enter the deadlock by making their flags true and spin lock by busy waiting in while loop.
- Bounded wait satisfies.

**135. (d)**

Number of cylinder = Number of tracks on single surface

$N_{\text{cylinder}} = N_{\text{track}}$

Disk capacity = Number of surface × number of tracks/surface × number of sector/track × sector size

= Number of surfaces × number of cylinders × number of sectors/track × sector size

Disk capacity can also computed from cylinders.

Disk capacity = number of cylinders × cylinder capacity

$\therefore$ All options are correct.

**136. (60)**

| Process | Currently Allocated | | | | Maximum | | | | Needed | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | U | V | W | X | U | V | W | X | U | V | W | X |
| A | 0 | 0 | 1 | 2 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 |
| B | 1 | 0 | 0 | 0 | 1 | 7 | 5 | 0 | 0 | 7 | 5 | 0 |
| C | 1 | 3 | 5 | 4 | 2 | 3 | 5 | 6 | 1 | 0 | 0 | 2 |
| D | 0 | 6 | 3 | 2 | 0 | 6 | 5 | 2 | 0 | 0 | 2 | 0 |
| E | 0 | 0 | 1 | 4 | 0 | 6 | 5 | 6 | 0 | 6 | 4 | 2 |

(i)

$$\begin{array}{rcccc l}
 & U & V & W & X & \\
\text{Available} = & 1 & 5 & 2 & 2 & \\
A \rightarrow & 0 & 0 & 1 & 2 & \\
\hline
 & 1 & 5 & 3 & 4 & \\
C \rightarrow & 1 & 3 & 5 & 4 & \text{(Here, D can also satisfy)} \\
\hline
 & 2 & 8 & 8 & 8 & \text{B or D or E can satisfy its need}
\end{array}$$

Starting with A and then C has 3! = 6 safe sequences.

$$\left.\begin{array}{l}
A, C, B, D, E \\
A, C, B, E, D \\
A, C, D, B, E \\
A, C, D, E, B \\
A, C, E, B, D \\
A, C, E, D, B
\end{array}\right\} \text{Six safe sequences.}$$

(ii)

$$\begin{array}{rl}
 & U\ V\ W\ X \\
\text{Available} = & (1\ 5\ 2\ 2) \\
D \rightarrow & (0\ 6\ 3\ 2) \\
\hline
 & (1\ 11\ 5\ 4)\ \ \text{A or B or C or E can satisfy its need}
\end{array}$$

∴ Starting with D process, 4! = 24 safe sequences.

(iii)

$$\begin{array}{rl}
 & U\ V\ W\ X \\
\text{Available} = & (1\ 5\ 2\ 2) \\
A \rightarrow & (0\ 0\ 1\ 2) \\
\hline
 & (1\ 5\ 3\ 4) \\
D \rightarrow & (0\ 6\ 3\ 2) \\
\hline
 & 1\ 11\ 6\ 6\ \ \text{B or C or E can satisfy its need}
\end{array}$$

∴ Starting with (A, D) there are 6 possible safe sequences.

(iv)

$$\begin{array}{rl}
 & U\ V\ W\ X \\
\text{Available} = & (1\ 5\ 2\ 2) \\
C \rightarrow & (1\ 3\ 5\ 4) \\
\hline
 & (2\ 7\ 7\ 6)\ \ \text{A or B or D or E can satisfy its need}
\end{array}$$

∴ Starting with C process, 4! = 24 safe sequences.

From (i), (ii), (iii) and (iv).

⇒ Total safe sequences = 6 + 6 + 24 + 24 = 60

**137. (b)**

- To get the minimum value of count.

| **Tally1( )** | **Tally2( )** | **Tally3( )** |
|---|---|---|
| Iteration 1. I: count = 0, $R_1$ = 0 | | |
| II: count = 0, $R_1$ = 0 | | |
| | Iteration 1. count = 1, $R_2$ = 1 | |
| | 2. count = 2, $R_2$ = 2 | |
| | 3. count = 3, $R_2$ = 3 | |
| | 4. count = 4, $R_2$ = 4 | |
| | 5. count = 5, $R_2$ = 5 | |
| | | Iteration 1. count = 6, $R_3$ = 6 |
| | | 2. count = 7, $R_3$ = 7 |
| | | 3. count = 8, $R_3$ = 8 |
| | | 4. count = 9, $R_3$ = 9 |
| Iteration 1. III: count = 1, $R_1$ = 1 | | |
| | | Iteration 5. I: count = 1, $R_3$ = 1 |
| | | II. count = 1, $R_3$ = 2 |
| Iteration 2. count = 2, $R_1$ = 2 | | |
| 3. count = 3, $R_1$ = 3 | | |
| 4. count = 4, $R_1$ = 4 | | |
| 5. count = 5, $R_1$ = 5 | | |
| | | Iteration 5. III: count = 2, $R_3$ = 2 |

$\downarrow$ Time

- To get the maximum value, execute $P_1$, $P_2$ and $P_3$ completely in sequence.

$\therefore$ Count = 15

**138. (d)**

$$\text{Page table size of 1st level} = \frac{2^{32}\,\text{B}}{2^{10}\,\text{B}} \times 4\text{B} = 2^{24}\ \text{B} > 1\ \text{KB}$$

$$\text{Page table size of 2nd level} = \frac{2^{24}\,\text{B}}{2^{10}\,\text{B}} \times 4\text{B} = 2^{16}\ \text{B} > 1\ \text{KB}$$

$$\text{Page table size of 3rd level} = \frac{2^{16}\,\text{B}}{2^{10}\,\text{B}} \times 4\text{B} = 2^{8}\ \text{B} < 1\ \text{KB}$$

So 3 level of page table required.

**139. (d)**

Set 3 contain $S \rightarrow \cdot$ as reduced item.

Follow (S) = { ), \$}

In row 3, entry for column ')' and '\$' will be "$r_b$".

$E_1 = r_b$, $E_2 = r_b$ $[\because\ b : S \rightarrow \varepsilon]$

In set 5, on S it goes to set 6.

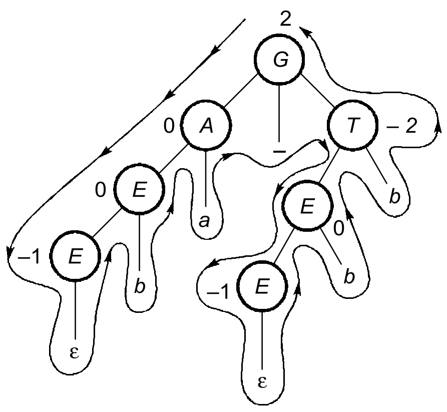
In row 5, entry for non-terminals 'S' is state 6.

$E_3 = 6$ $[\because\ (5) \xrightarrow{S} (6)]$

$E_1 = r_b$, $E_2 = r_b$ and $E_3 = 6$

**140. (2)**

**Input:** ba–bb



In above dependency graph, 2 is the value at G after the evaluation.

**141. (d)**

Set 1 has RR conflict.

$A \rightarrow a., \{b\}$

$B \rightarrow ba., \{b, c\}$

$\{b\} \cap \{b, c\} \neq \phi$

$\therefore$ Grammar produces RR conflict for CLR (1).

**142. (b)**

$d = R_1 = t_2 = R_1 = R_1 + R_2 = R_1 + t_1 = a + t_1 = a + R_1 = a + R_1 * R_2 = a + b * c$

**143. (2)**

$$\left.\begin{aligned} t_1 &= t+e \\ t_2 &= g+a \\ t_3 &= t_1 * t_2 \\ t_4 &= t_2 + t_2 \\ t_5 &= t_4 + t_3 \end{aligned}\right\} \Rightarrow \begin{aligned} t_1 &= t+e \\ t_2 &= g+a \\ t_1 &= t_1 * t_2 \\ t_2 &= t_2 + t_2 \\ t_1 &= t_1 + t_2 \end{aligned}$$

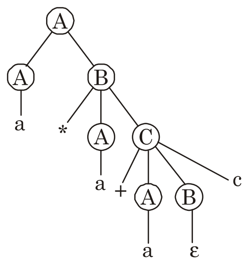
Above code represents the following expression:

$((g + a) + (g + a)) + ((t + e) * (g + a))$

∴ Only two temporary variables are used.

**144. (7)**

String = "a ∗ a + ac"



# of reductions = # of intermediate nodes = # of substitutions = 7 [reduction order in LR parser must be reverse of RMD]

**145. (b)**

**Viable prefixes:** Combination of non-terminal and terminal which can be in the stack during parsing is called viable prefix.

A handle X→ b is available. So whenever terminal b is encounter it is popped out before inserting new element into the stack.

Option (a) is wrong because before popping 'b' element 'aab' is inserted.

Option (b) is correct after 'b' no element has inserted.

So, option (c) and (d) also incorrect.

**146. (c)**

Any symbols $, ', @ like that gives lexical error in C if put any where outside of string and the comment lines. They can not pass the lexer because lexer can not recognize them as a valid token.

- Program 1 gives lexical error because "@_" written outside comment.
- Program 2 having no lexical error but producing syntax error by syntax analyzer.
- Program 3 gives lexical error closing comment lines */ is missing.
- Program 4 gives lexical error because integer has been assigned an invalid octal number "09".

**147. (b)**

As we know,

$A \rightarrow A\alpha \mid B$

After eliminating LR

$A \rightarrow BR$

$R \rightarrow \alpha R \mid \in$

In the given case

$$\alpha = Y\ \{X.x = f(X.x,\ Y.y)\}$$

Final production after eliminating will be

$X \rightarrow Z\ \{X.x = g(Z.z)\}\ R$

$R \rightarrow Y\ \{R.x = f(X.x,\ Y.y)\}\ R$

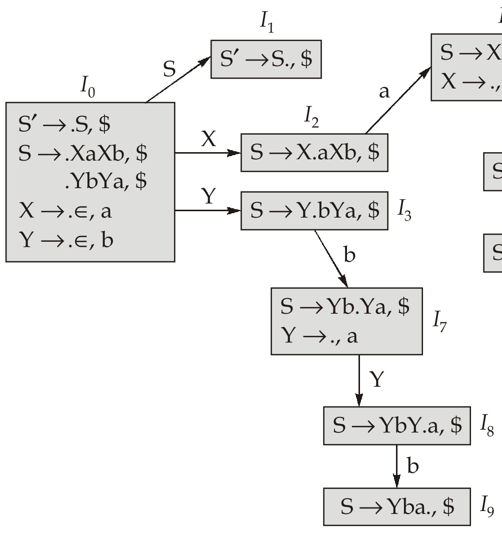
$R \rightarrow \in$

So, option (b) is correct.

**148. (b)**

$$\begin{aligned} \text{First (S)} &= \{a, b\} \\ \text{First (X)} &= \{\in\};\ \text{FOLLOW (X)} = \{a, b\} \\ \text{First (Y)} &= \{\in\};\ \text{FOLLOW (Y)} = \{a, b\} \end{aligned}$$

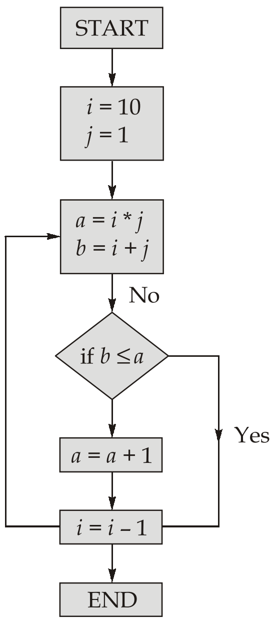
→ Grammar is LL(1).



Grammar is not SLR(1) because of conflict at the state $I_0$. Hence it will not be LR(0) also. The above grammar is LR(1).

Hence option (b) is correct.

**149.** **(15)**



Number of nodes X = 7
Number of edges Y = 8
= X + Y = 15

**150.** **(a, b, c)**
A loader is the part of an operating system that is responsible for loading programs and libraries.

■■■■