TECHNICAL INTERVIEW QUESTIONS

for

CS/IT Students

FIRST EDITION

Collected and Set by

ARUNANAND T A

Exclusively Prepared for

COLLEGE OF ENGINEERING CHENGANNUR

Alappuzha District, Kerala, India - 689121

DEPARTMENT OF COMPUTER ENGINEERING

TECHNICAL INTERVIEW QUESTIONS

(for CS and IT students)

ABOUT THE AUTHOR

Arunanand T A was born on 11th April, 1989 at Nayarambalam, a small village in the island of Vypin, Ernakulam Distirct, Kerala. He did his secondary and higher secondary schooling at BVHS, Nayarambalam and SMHSS, Cherai respectively. After taking graduation in B.Tech Computer Science and Engineering from College of Engineering Chengannur, Kerala (*affiliated to Cochin University of Science and Technology*) he's currently preparing for his Masters program in Engineering.

His father is Mr. Anandan T N and mother Ms. Kumari Anandan. He's currently residing at his native place referred to below.

Address: Thayyathazhath H, Nayarambalam P O, Ernakulam, Kerala, India – 682509.

Contact No: +91 – 9495 087 682.

E-mail: arunta007@gmail.com

More about the Author: http://www.arunanand.com | http://www.cecblog.com

This page is intentionally left blank

CONTENTS

PREFACE

This book is intended to help the students from technical backgrounds perform well in their technical interviews. Questions presented herein are from author's personal experience or collected his friends, teachers, magazines, books or other online resources.

The books aims at providing a coherent approach in technical subjects; however, apart from technical topics, some headlines of general interest for technical students have also been included as annexure at the end of this publication.

A Word of Thanks

I would like to thank all those who have helped me in preparing this document, especially my teachers and friends at College of Engineering Chengannur, Kerala and ACE Engineering Academy, Hyderabad for assisting me. Special thanks to Ms. Nisha Kuruvilla, Asst. Professor in Dept. of Electronics Engineering and the Placement Officer at College of Engineering Chengannur, who encouraged me in writing this booklet.

I extend my thanks to all those who have shown a good heart towards the attempt, especially to Mr. Fredie Thomas, Mr. Viswajith V Kurup, Mr. Sanoop Thomas and Mr. Abin John George.

Suggestions Invited

Suggestions and inclusions are invited from teachers, experts, enthusiasts and students. Kindly send me your comments and new questions to: ***arunta007 [at] gmail [dot] com***

Disclaimer

This document contains questions collected from various students, teachers, magazines, online resources and/or set by the author. However, the author will not bear any responsibility over any kind of discrepancies that might have crept in while setting the document.

Ernakulam,

25-Nov-2010. **ARUNANAND T A**

TO STUDENTS

First of all, let me thank you for having a look at this booklet. As you would probably understand, this book contains some *tentative or likely or, rather, potential* questions for a technical interview a CS/IT student may face. I would like to emphasize on the word *potential* as there is no sure about whether any of the questions presented herein will be asked in an interview or not. It depends on the domain for which interview takes place, and the level of knowledge the company expects from you. You may, perhaps, face not even a single question from this collection, and sometimes, most of your interview may be on the topics covered herein.

Hence, my sincere advice to you is that you consider this booklet as a guideline, or as a model, which may help you understand how to prepare for a technical interview. Remember, Google and Wikipedia are your best friends, always! However, never do avoid text books. I have some suggestions in this regard, which I have kept in the next page.

And I am sorry to inform you that I have not added all the solutions to the questions. It’s the sole responsibility of the aspirants to find them out. However, to your advantage, hints to some vital questions have been provided wherever necessary.

Wish you all the best in your knowledge and job hunt!

Ernakulam,

25-Nov-2010. **ARUNANAND T A**

SUGGESTED READINGS

Here are some suggested readings an undergraduate student of CS/IT should refer to. Nevertheless, they may, perhaps, not be the best text available in the market. This is simply from my experience.

Some text books are written by Indian author, some by foreign authors and some by both. It’s also to be noted that these are the widely used text books for competitive examinations and technical interviews.

- **Mathematics**
  “ Higher Engineering Mathematics” by B.S.Grewal

- **Automata Theory and Computation (Theory of Computation)**

  “Formal Languages and Automata theory” , J.D.Ullman

- **Design and Analysis of Algorithms**

  “Introduction to algorithms” – Cormen etal (CLRS)
  “Computer Algorithms” – Horowitz and Sahani

- **Digital Logic**

  “Digital Logic circuits and Design” by Morris Mano

- **Computer Architecture and Organization**

  “Computer Organisation” by Morris Mano
  ” Computer Architecture” by Briggs

- **Data structures**

“Data structures” Schaumm’s outline series
“Data structures in PASCAL” by Horowitz and Sahani
“Data structures and Algorithms” by Weiss
“Introduction to Algorithms” – Cormen

- **C/C++ Programming**

  “Programming with C” – Byron Gottfried , Schaumm’s outline series
  “Principles of Programming Languages” by Robert W Sebesta , Addison Wesley
  “Programming with C++” – Balaguruswamy

- **Compiler Design**

  “Principles of Compiler Design” , Aho , Ullman etal .
  “Systems Programming” by John . J . Donovan

- **Operating Systems**

  “Operating system concepts” by Abraham Silberschatz and Peter Galvin

- **Databases**

  “Principles of Database Systems” – J.D.Ullman
  “Database system concepts” – Silberschatz , Korth , Sudarshan
  “Database systems” – C.J.Date

- **Computer Networks**

  “Computer Networks” – Tenenbaum
  “Data communications and Networking” – William Stallings

- **Some more important books :**
  “Multiple choice questions” – Timothy . J . Williams , TMH publications
  “Gate Question Papers ” – G.K.Publishers.
  “Aptitude – Verbal and Non-Verbal” by Aggarwal

Dedicated to my friends at CEC…

Without whom

I couldn’t have completed this work

and

I wouldn’t have understood the real joy of life.

CHAPTER 1

INTRODUCTION

POTENTIAL TECHNICAL INTERVIEW QUESTIONS

Following are some potential questions *(from core subjects only)* those can be expected in the CS/IT Technical Interviews. Questions are classified subject-wise. These questions are collected from various students who have undergone such interviews, internet, some magazines and M.Tech/ME/MS students who had to face such questions during their admission to institutions of national importance.

N.B: Some Electronics subjects that still have strong relation with CS/IT branches, like ***Digital Electronics, Microprocessors, Data Communication,*** *etc. are not included in this document. Students may expect questions from such subjects also.*

Also, students are advised to get themselves updated with recent technologies released in the market, some key technologies like system architectures, distributed/cloud/grid networking/computing, and general questions like Von-Numann Architecture, Moore's Law, etc.

CHAPTER II

DATA STRUCTURES

(Students are supposed to be well aware of the programming concepts using Arrays, Linked Lists, Trees and Graphs. It's likely that the interviewer may provide you with a scrap of paper and ask to write down certain program modules. This is applicable to all those subjects which involves programming portions)

1. Why do we require data structures?
2. What is the difference between a physical structure and a data structure?
3. If I say that linked list is a linear data structure, will you assent to me?
4. What is the difference between a B Tree and B+ Trees? Have they got any similarities?
5. What do you know about AVL rotation?
6. What is the last thing you learned about data structures from a book, magazine or web site?
7. You might have used 2-D arrays while programming. But you know that our main memory is of one dimension. How is it possible to establish this?
8. What is FIFO?
9. Can you differentiate between Array and Linked List? Which one is linear and what are the advantages being linear?
10. What do you mean by a circular queue? Can you suggest a situation where we can use this?
11. What do you mean by rear and front end of a queue?
12. *(This question is normally a continuation of the above question)*What if I insert at the rear end and delete from the front end, what will happen?
13. Have you ever used priority queues? Where do we use them in practical applications?
14. How can a stack be implemented? Which one do you prefer? Why?
15. What would you imagine the problems that could happen if the data structures didn't exist?

16. How do you explain what a data structure means, to a 12-year old nephew of yours?

17. Suppose you are a teacher and a student comes to you with his ambiguities between databases and data structures. How could you make him understand these two?

18. What are activation trees? What is its use?

19. Which data structure is used for better performances under insertion and deletion on a dynamic basis?

20. What is the Worst Case of “Merge Sort”?

21. Where are linked lists are used?

22. Could you please explain Dijkstra’s Algorithm in brief?

23. Is double link list a linear data structure? Why?

****

CHAPTER III

DISCRETE MATHEMATICS AND COMPUTATIONAL STRUCTURES

1. Do you think Dijisktra's Algorithms is flawless to find out the shortest path?
2. Are trees and graphs are relevant in computing? Where?
3. What do you mean by an equivalence relation?
4. Explain 'Pigeon Hole Principle'?
5. What is a spanning tree?
6. What's the difference between a *complete binary tree* and *full binary tree*?
7. What do you mean by a 'regular graph'?
8. What is the difference between a graph and a tree?
9. Do you know about the Traveling Salesman Problem?
10. Have you heard about Graph Coloring Problem with Minimal number of Colours?
11.

****

CHAPTER IV

COMPUTER ARCHITECTURE AND ORGANIZATION & SYSTEM PROGRAMMING

1. What’s the importance of Subroutines in Programming? Which data structure is widely used in Subroutine concept?
2. What do you mean by Software interrupts?
3. What is pseudo-parallelism?
4. What are the functions of Program Counter and Stack Pointer?
5. What dimension of memory is RAM?
6. Why do we use cache?
7. Define DMA?
8. What is Process Interleaving?
9. Tell me how you will replace a BIOS battery?
10. Differentiate RISC and CISC?
11. Out of Intel Core 2 duo and Dual Core, which one is more efficient?
12. Which is the newest series of processors from Intel, available for the common public?
13. What are DDR2 and DDR3?
14. Can you explain how the execution of an instruction is done, in detail?
15. Is it possible for a program to get preempted *amidst* executing a C statement ***a=b+c;*** in a block of programs? *(Ans: Yes, actually all instructions are executed in equivalent assembly language codes. The assembly language code of the above single C statement contains more than one instruction. Hence, it is possible for the statement to get preempted amidst the execution of the given statement)*
16. What is the difference between static and dynamic memories?
17. What do you mean by vectored interrupts?
18. Which one – linker or loader – is *essential* for a computer? *(Ans: loader. It’s essential that data to be processed (OS+Data) to the functional memory unit. For this loader is a must. Linker’s idea is different from this basic need)*

19. What do you mean by relocatability of a program?
20. What do you mean by a macro?
21. What do you mean by pipelining?
22. Explain the concept of virtual memory?
23. What is write back and write through caches?
24. What is miss penalty and give your own ideas to eliminate it.
25. What is aliasing?
26. What is the difference between a latch and a flip flop?
27. What is the race around condition? How can it be overcome?
28. What is meant by Snooping Cache?
29. Explain how to convert a number into its 2’s complement and convert back.
30. Explain about various hazards?

Note: *Prepare yourself with knowledge in Number Systems and Conversions, Data Structures, Interrupts, etc. At least, they might help you in the written session, if any.*

****

CHAPTER V

SOFTWARE ENGINEERING

1. What do you mean by CMM level? Do you know what CMM level we possess? *(question asked by Infosys)*
2. Suppose you are automating the library of your college. Which model would you adopt?
3. Which of the three – design, coding, implementation and maintenance – do you think is of utmost importance?
4. Can you tell me something about COCOMO?
5. Suppose in your company there was a dispute in terms of technical enhancements of the firm. There are around 25 employees taking part in the dispute. Which method would you rely on to sort out the dispute? *(student is supposed to have some basic knowledge in activities like brain storming, questionnaire, etc. It's suggested that they go through B.Tech First year TCSS' relevant topics)*
6. Which process model do you think is the best? Justify *(do not blankly say one model is the best; be smart and diplomatic enough to state that each model finds its best role in various types of software designs. Cite some examples and justify)*
7. Why do you consider an SRS to be of utmost important document in a software project?
8. What does an ER-diagram represent?
9. Which are the different phases in a software life cycle?
10. What do you mean by a Good Software Design?
11. How do you search and find requirements? What are possible sources?
12. What do you think is the limitation of an E-R diagram?
13. Why are analysis and testing phases very important?
14. Why should you study a theory-based subject like *Software Engineering*? Is it that much important for a *Software Engineer on field*?
15. What do you mean by testing? What is black box testing?
16. What are the objectives of debugging?
17. What is an SRS?

18. How many of the three variables scope, time and cost can be fixed by the customer? Why?

19. Do you think whether documentation is an important activity during a software project?

****

OPERATING SYSTEMS

1. Can you explain the working of ***fork()*** system call?
2. What is the concept of preemption in scheduling?
3. What is the importance of real time systems?
4. Which CPU scheduling algorithm do you think is the best? Why?
5. What do you know about file systems in UNIX and Windows?
6. What does NT stands for in Windows NT?
7. Expand GNU?
8. Tell me something about paging?
9. What are real time systems?
10. What is the concept of Master and Slave at the time of booting?
11. Expand BIOS? What do you know about recent advancements/substitute for BIOS? *(Hint: UEFI. Students are supposed to keep track of such latest advancements in the field, through dailies, internet and magazines. These kinds of hints are not usually given by the interviewer, but sometimes he may try to help you in case you do not answer the question)*
12. Are semaphores, monitors, etc. important concept in Kernel Programming? Why?
13. What if I say, as the degree of multi programming increases, CPU usage also increases – do you agree? *(Ans: No. It increases first, then steady and then decreases)*
14. What are the advantages of threads over processes? Have they got any disadvantages?
15. What is an Operating System?
16. What do you mean by deadlock? Is deadlock possible without the concept of shared memory?
17. What do you know about Process Spawning?
18. Why do we call thread a ‘light-weight process’?
19. What are the conditions for deadlocks?
20. What is meant by Busy Waiting?
21. What does PCB contain, in regard of a process?

22. What is process interleaving?
23. Can you explain ‘Banker’s Algorithm’?
24. Tell me something about disk fragmentation?
25. What do you know about starvation?
26. Is ‘directory’ a file? Explain?
27. Why do we use the concept of *paging*?
28. Tell me something about Mutual Exclusion?
29. Why page replacement algorithms are required?
30. Can you compare FAT-32 and NTFS? What does this 32 mean?
31. What are the functions of Interrupts?
32. What is done during a booting process?
33. What do you think about dynamic memory allocation?
34. Why do we use virtual memory?
35. What’s the difference between a system call and a system program?
36. Why in most USB Drives FAT is preferred to NTFS?
37. Can you suggest some differences between *“Physical Address Space”* and *“Logical Address Space”*?
38. What is a DRAM?
39. Explain Deadlocks?
40. Differentiate *Starvation* and *Aging*?
41. Which is the latest mobile platform introduced by Google?

****

DATABASE MANAGEMENT

1. How will you make your school-going brother understand the difference between the terms ‘Data Structure’ and ‘Database’?
2. Expand and explain SQL?
3. Can you write an SQL query for extracting *all “name” of people in a table “T1” which contain the substring “sa” and having “age” between 35 and 45*? (Students are advised to have a basic knowledge of SQL commands for various basic operations on tables)
4. Why do you say that normalization is needed? How many levels of normal forms exist?
5. What have you got to tell me about data warehousing?
6. What do you mean by a transaction?
7. Tell me something about ACID properties?
8. Differentiate between a *‘where’* clause and *‘having’* clause in SQL?
9. Can you suggest a situation in our day-to-day life where the concept of Transaction Ordering Management is of utmost importance?
10. What do you mean by the statement ‘a transaction is committed’?
11. Why do we sometimes need to resort to *denormalization*?
12. What is atomicity?
13. Differentiate between a *Candidate Key*, a *Foreign Key* and a *Primary Key*?
14. What do you mean by concurrency control? Why is it needed?
15. Differentiate between DBMS and Database System?
16. Give me an example of popular Database System?
17. What is Functional Dependency?
18. Tell me something about *“triggers”* in data bases?
19. What are the different phases of transaction?
20. Tell me two advantages of using Data Bases?

21. Write an SQL query to return the name of the fourth topper among the students of a class? (Table Name = ***student*** ; Distinction Identifier = ***rank*** ; Name of the Student = ***student_name***)

22. Give the SQL query to return the names of the customers who have bought both the items P1 and P2? (Table Name = ***customer*** ; Item Name = ***item_name*** ; Customer Name = ***customer_name***)?

23. Suppose there is a table with the following entries. I want to delete all the repetitive rows, retaining only one copy. (There's no key field concept applied to the table) Write an SQL query.

id	***name***	***age***	***sex***
101	*ABC*	*15*	*M*
102	*PQR*	*16*	*F*
101	*ABC*	*15*	*F*
103	*PQR*	*14*	*F*
101	*ABC*	*15*	*M*
101	*ABC*	*15*	*F*

24. What is the major difference between a DBMS and an RDBMS?

25.

****

AUTOMATA THEORY AND LANGUAGES

1. Where is Myhill-Nerode theorem is used? Could you please explain the algorithm?
2. If I say PDA=FA+2 Stacks, what is FA+3 stacks?
3. Every regular language is context free. Do you agree? Why?
4. Can you compare a PDA with a microcomputer?
5. If I say DFAs are more realistic than NFAs, will you agree with me?
6. You know that the closure property exists under concatenation of two regular languages. Now explain: $\boldsymbol{a^n}$ is regular and $\boldsymbol{b^n}$ is regular. But why $\boldsymbol{a^n b^n}$ is not regular?
7. If I say some language is context free, what does it imply?
8. What would you call the ***State Diagram of a Process*** – an NFA or a DFA?
9. Can you check and say $L = \{a^{n^2}, where\ n = 1,2,3, \ldots .\}$ is regular or not?
10. What is the difference between regular grammar and algebraic grammar? *(Question asked to GATE 2010 Topper at IISc, Bangalore during his admission process)*
11. When do you call a problem decidable?
12. What do you mean by saying that a language is LL(1)?

****

ALGORITHMS & COMPUTER PROGRAMMING LANGUAGES

(To ask to write program modules in C/C++, and sometimes in JAVA too, is quite a common trend in interviews. Be prepared to face such questions and write down some famous modules)

1. Write a C Program module for calculating the factorial of a number *using/without using** recursion? *(*Sometimes with recursion and sometimes without recursion are being asked)*
2. Write a C Program module for finding the first *n* terms in a Fibonacci Series *using/without using* recursion?
3. What are time complexity and space complexity?
4. What is the use of Recursion? Is it compiler-friendly?
5. How do you find out if a number is a power of 2?
6. Is pointer a variable?
7. Why java does not contain pointers?
8. What are constructors and destructors? Then, what do you mean by Virtual Destructors?
9. What do you mean by *malloc()* in C? Then, what is *dealloc()*? *(Beware, the interviewer may mislead you by invalid keywords or so)*
10. Write a C Program module to reverse a string, without using *string.h* library functions?
11. What is cascading of operator?
12. What are the basic features of OOP? *(Students are advised to study the details of all the basic features of OOP, as this is a very much repeated question in almost all technical interviews. Hence, in this document, further questions regarding each feature of OOP is not included – it's to be learned as such)*
13. Differentiate between POP and OOP?
14. What are the differences between *Syntax, Logical* and *Run Time errors* in C*?*
15. How to convert decimal number to binary? *(You need not write the code module, but explain the algorithm)*
16. Why is the complexity of an algorithm to be determined?
17. How the compiler make difference between C and C++?
18. What do you mean by the robustness of a language?

19. What’s the use of Scope resolution operator?

20. What is the general strategy for Markov Algorithm?

21. How many temporary variables is essential for swapping two integer variables? *(Usually, we use one temporary variable; but it’s not essential. Hence the answer is ZERO. Find out how you can swap without using temporary variables)*

22. What is the difference between = and == in C?

23. How many but combinations are possible in a byte?

24. What are function overloading and operator overloading? Which concept of OOP do they comply with?

25. What are *friend* functions in Class?

26. Why do we declare some functions are *virtual* in Class?

27. How do you explain to a kid the concept of Classes and Objects? *-OR-* Can you find an analogy of Classes and Objects in the real world?

28. What shall be the output of the following code:

    ```
    int i=50;

    printf(“%d,%d”,i,i++);
    ```

    (Yes, sometimes, this also has happened. When they feel that the candidate is appealing, they will give you a small program code in writing, which be tricky in nature. In the above question, the answer is 51,50 which may be wrongly-attempted by the candidates)

29. Write a C Program module to find the numbers between 10 and 99 where the number is thrice the sum of its digits?

30. Find the GCD of two numbers using C/C++?

31. What do you know about *Type Casting*?

32.

33. What’s the use of pointers? Any disadvantages?

34. Why is it suggested not to use malloc() or calloc() in C++ for memory allocation?

35. Can array name itself be used as a pointer? How?

36. What is *Virtual Destructor*?

37. What is the difference between basic building blocks in C and C++?

38. Can a pointer be pointed to by another pointer?

39. What if I say an *1-D* array of pointer is enough to point to a *2-D* array of numbers, will you agree with me? Explain.

40. Can you give me an example for a tertiary operand used in C?

41. Why we follow top down approach in C-language and bottom up approach in C++? Explain in detail.

42. Write a function in C/C++ to reverse the words in the sentence, for example "I am from CEC" becomes "CEC from am I"

43. Can you give me an instance where we may need to resort to 3-D array representation, instead of confining to 2-D?

****

COMPILER DESIGN

1. Can you differentiate between ‘syntax analysis’ and ‘semantic analysis’ during compiler design?
2. In simple words, what shall be the basic difference between top-down and bottom-up parsing?
3. Can you say something about type checking and its importance?
4. What is the need for Symbol Table?
5. What is the fundamental difference between a ‘compiler’ and an ‘Interpreter’?
6. Which of the two – linker, loader – is essential in a microcomputer? Can you justify your answer?
7. How is Code optimization done using DAG?

****

COMPUTER NETWORKS, NETWORK SECURITY AND WEB TECHNOLOGY

1. What does a firewall do?
2. What are public key and private key cryptographies?
3. What do you know about Digital Signatures?
4. Can you expand VIRUS?
5. Expand PING and explain how it works?
6. What bit addressing system does Ethernet have?
7. Can you differentiate between Security and Protection?
8. Can you differentiate between a router, abridge, a repeater, a gateway, a hub and a switch? *(This is a very common question. Two or three of these may be asked to be differentiated from one another)*
9. How many hosts are possible in a Class C IP Address?
10. Tell me something in brief about Classes of IP addresses?
11. What's the advantage of IPv6 over IPv4?
12. Some people say that Subnetting and Supernetting are two important concepts of networking. Do you agree? Tell me something interesting about them.
13. Why is OSI layered?
14. Tell me something about ISO/OSI?
15. Why do we need Congestion Control techniques?
16. Tell me something about fragmentation of data packets?
17. Out of the two – TCP/IP and UDP – which is connectionless protocol. Why? Compare.
18. Which protocol is used in Video Streaming – TCP/IP or UDP?
19. Tell me something about Client-Server and Peer-to-Peer architectures?
20. Have you heard about torrents? How does a torrent file works?

21. Which Flip Flop is most used in data communication over networks? *(Ans: D Flip Flop, as it gives delay. In most of the ports, this flip flop is used, as the sender and client have to be synchronized. It's also useful when the receiver is slow compared to the sender)*

22. Tell me something about Piggy-Backing?

23. Tell me about HTTP, SMTP, SNMP, FTP, ICMP…..? (*Students may expect questions on any one of such layer protocols)*

24. What is RFC?

25. Which protocol is called 'watch dog'?

26. Tell me something about POP and SMTP in E-mail services? Give an example of POP/SMTP client? *(E.g: MS Outlook, Mozilla Thunderbird, etc.)*

27. What are the advantages of IMAP over POP?

28. What is Loop Back address? Explain?

29. What is ARP?

Above are frequently asked Computer Network questions in Technical Interviews. Some other typical Computer Network questions, based on the CUSAT syllabus, are added below (collected from Internet sources). This is because Computer Network questions have much and vital importance in technical interviews and other competitive examinations.

- Can you define a protocol?
- Is it compulsory that compression, encryption and translation functions will be used during communication?
- What's the concept of Simplex, Half Duplex and Full Duplex dialogs?
- Can you explain what an IP address is?
- How many IP addresses can be there in IPv4?
- Can you explain the concept of Unicast IP address?
- Can you explain the concept of IP multicasting or multicast IP address?
- What is Broadcasting in terms of Computer Networks?
- What are the IP address ranges for public and private IP address?
- If the host has the subnet ID why do we need a subnet mask?

- How is network address calculated from the subnet?
- What is the advantage of using classless addressing over classful addressing scheme?
- Why do we need to subtract two from number of hosts in all IP Classes, to get the effective number of hosts?
- Can you explain the concept of VLSM?
- Can you explain IP packet in detail?
- Can you explain the concept of DPU, Segments, Datagram, Frame and packet?
- What is IP datagram fragmentation and MTU?
- Larger the IP datagram less the overhead, is it true or false?
- What is the minimum size of MTU bytes?
- Can you explain how optimal MTU size is calculated?
- How does the IP message finally reassemble?
- On what layers do router, switches, bridges and hubs operate?
- What are the basic components of the router?
- Can you explain the concept of TTL?
- What is the concept of ICMP packets?
- Can you explain the concept of NAT?
- How is NAT implemented?
- What is the concept of routing tables?
- What is the use of route print?
- Can you explain how a routing table looks like?
- Can you explain the concept of static and dynamic routing?
- When to use Static routes and dynamic routes?
- What is the difference between Source-Routing and Router-Routing?
- Can you explain static default routes?
- What is the advantage of using Static default routes?
- Why do workstations have route tables?

- Can you explain the concept of routing protocol?
- What activities does routing protocol perform?
- What metrics are used by routing protocols to determine the best path?
- Can you explain what is interior and exterior routing protocols?
- Can you explain the concept of intradomain and interdomain routing protocols?
- Can you explain the concept of internet work and intranet work routing protocols?
- Which method does routing protocol use to determine shortest path?
- What is distance vector routing protocol?
- How do routers share information in Distance Vector routing?
- Can you explain the count-to- infinity problem in distance vector?
- How is metric or the cost calculated for Distance Vector routing protocol?
- Can you explain how Link-State routing protocols work?
- Can you explain the concept of broad cast and multi-cast?
- What's the difference between distance vector and link-state protocol?
- Can you explain Active and Passive route?
- Can you explain OSPF?
- How does OSPF populate route table?
- What are the different tables in OSPF?
- Can you explain different areas in OSPF?
- Can you explain different router types in OSPF?
- Can you explain Designated Router and Backup designated router?
- Can you explain different router states in OSPF?
- Can you explain different OSPF packet types?
- How SPF algorithm does the route determination?
- Can you explain autonomous system?
- What are different types of dynamic protocols?
- Can you explain autonomous numbers in EGP?

- What is BGP?
- What is the concept of BGP speakers and Peers?
- What is EBGP and IBGP?
- What is RIB?
- Can you explain the concept of BGP confederations?
- What are BGP path attributes?
- How are routing neighbors discovered in BGP?
- Can you explain how BGP does the decision process?
- Can you define what a FIREWALL is?
- What are the different types of firewalls?
- Can you explain packet filtering firewall?
- Can you explain circuit level gateway?
- Can you explain stateful inspection?
- What is Application Gateway?
- Is NAT a firewall?
- Are personal firewall actually firewalls?
- What is the meaning of bastion host?
- What are the different types of firewall architectures?
- Can you explain dual home architecture?
- Can you explain screened host architecture?
- Can you explain screened subnet architecture?
- What is IP spoofing and how can it be prevented?
- With which firewall have you worked with? How was the experience?
- Can you explain the difference between trusted and untrusted networks?
- What’s the difference between Symmetric and Asymmetric cryptosystem?
- What are the different symmetric algorithms?

- What are the disadvantages of symmetric algorithms?
- What are the different asymmetric algorithms?
- What is a digital certificate?
- Can you explain what IPSec is?
- Can you give an overview of various components in IPSec?
- Can you explain in a generic manner the packet of IPSec?
- Can you describe the Authentication Header (AH) Protocol?
- What do you mean by SSL? Why is it used?
- Can you explain what the use of IGMP Protocol is?
- How do you continuously ping an IP Address?
- Can you explain RPC (Remote Procedure Calls)?
- Can you explain TCP IP Protocol?
- Can you explain the architecture of TCP IP Protocol?
- Can you explain TCP header in detail?
- Can you explain the concept of CDMA?
- Can you explain the concept of DHCP?
- How does DHCP work?
- How can we configure DHCP?
- What is DNS?
- What is a difference between a domain and workgroup?

****

ANNEXURE V (B)

PREPARING FOR A TECHNICAL INTERVIEW

1. **Practice using the same medium (e.g. paper and pencil) and time limits (e.g. 30 minutes) as the real interview.**

   Google and Microsoft both use whiteboard coding questions, yet often candidates practice by coding alone at home on a computer with a compiler. During the actual interview, they stand at the whiteboard and forget how to initialize an array, without their trusty syntax highlighter. Or they are so nervous having another person watch them that they panic and can't think straight.

   In real life, if you plan to swim the English Channel, would you limit your practice to laps at the local swimming pool? No, you would go test out the ocean waves, the salt water. Do the same here.

   Ask your recruiter the format of the interview and any coding questions. If the company gives the candidates an hour alone in a room with an editor and no compiler, practice that at home. If the company does whiteboard questions with an interviewer watching you, ask an engineer friend to be your mock interviewer. It's fine if the friend is a less experienced engineer than you -- they'll still bring out your nervousness about making mistakes in front of others, so you can practice getting used to that.

2. **During the interview, don't obsess over little mistakes that happen.**

   On more than one occasion, when I gave a star candidate a coding question, he zeroed in on the most optimally performant solution, identified the boundary cases, and began writing well-designed code. Midway through the problem, he makes a little error -- getting the order of operations wrong on the first try, or having an off-by-1 error, or forgetting to declare a variable.

   When I point it out, the candidate responds with horror and then becomes so nervous that it impacts his performance during the rest of the interview.

   The fear is unfounded. An awesome candidate making a little error is like a concert violinist playing a challenging Brahms concerto and hitting two wrong notes. Sure, the audience could tell that he made mistakes, but they don't get confused as to whether he's actually at Twinkle-Twinkle-Little-Star level.

   Even if you completely bomb one question, many interviewers ask you multiple questions and will forgive a single mishap. Even bombing an entire interview is recoverable if the other interviews go well.

Recently one of my coworkers (a tech lead for another project) interviewed a candidate and was very curt because he found the candidate's communication style irritating. The candidate proved himself during the interview, and the tech lead ended up being the strongest proponent for this candidate. He advocated harder for that candidate than he has for anyone else in a year.

When things don't go well, just keep at it and don't give up hope.

3. **Don't be rude to your interviewer.**

This should be obvious, but I have been surprised. One engineering candidate said to me, "Wow, I can't believe you're really my interviewer! You look so young!! I thought you were 18! Once you told me your credentials, I understand now, but at first I thought, 'This person is interviewing me?!?!'"

That wasn't very graceful.

Other things that I recommend against saying:

- "You're really my interviewer? You look so old!"
- "You're really my interviewer? You look so fat!"

Another time, the candidate's cell phone rang 15 minutes into the interview. She let it go, and we were both distracted by it ringing for the next 20 seconds. 5 minutes later, it happened again. Another 5 minutes later, it rang for a third time.

She finally reached for her purse and fumbled inside it for the phone. "It's about time", I thought, "she should've turned it off before coming in here." She dug the phone out of the purse and then proceeded to *take the phone call* right there in the middle of the interview.

The only justification is if there is a family emergency, and in that case, warn your interviewer explicitly at the start of the interview.

4. **Don't hijack the interview.**

I've had a couple of candidates who came into the interview with the mindset that they MUST tell me all about their recent project Zoolander. I start the interview and they break in with, "I want to tell you about Zoolander. 10 years ago, this project started as a side feature..." and then go on for 5 minutes without taking a breath.

Sometimes they decide that they must tell every interviewer about Zoolander, repeating the same description over and over during the day.

Your interviewer has specific questions that they need to get through. If you hijack the interview, they may not have enough data from their own questions to be able to endorse your hiring. They may also think that you would be difficult to work with.

If you really want to talk about a project, ask your interviewer, "I think project Zoolander really shows off my abilities. Can you or another interviewer fit in 10 minutes for me to explain it?" The interviewer can then refit their plan for the interview, instead of suddenly having their schedule be shanghaied.

5. **When answering questions expecting a specific answer, give a high-level summary first.**

   Sometimes I ask a question expecting a short answer, "How many people worked with you on project Zoolander?" The candidate then gives me an audiobook, "Well, there was Jimmy -- he did the UI and I had to mentor him quite a bit on it. Then there was Mary who ran the backend servers. She worked remotely from Pennsylvania. Two years later, we got another backend person David..."

   Three minutes later, the candidate is still talking, and I still don't know the answer of how many people worked on the project.

   Give an answer first, and then expound. "There were 3 when I joined, and 12 when I left. First there was Jimmy ..."

   Better yet, give the answer and *offer to expound*. "There were 3 when I joined, and 12 when I left. Would you like me to tell you what each one did?"

6. **(Not as important) Wear something comfortable to your interview. Business casual is the most typical.**

   People sometimes wonder how they should dress. The most important thing is that you feel comfortable. If you still want a recommendation, I say a button-down shirt or even a T-shirt. A suit can come off as too formal in some companies (e.g. Google).

   This point is not as important, because people won't really care. You should ask your recruiter about what to wear, since this differs by country and East Coast / West Coast. A company like Google is more casual, so if you come in a three-piece suit, your interviewers may raise an eyebrow. If you've got the goods in terms of engineering skills, it's not a dealbreaker though. One candidate came to an interview wearing a gothic mesh shirt with holes through which his nipples were clearly visible. He still got the job. (I don't recommend taking this risk.)

A final story

I'd like to leave you with a story of an unfortunate interview. Draw hope that no matter how your interview goes, you will likely be more lucky than this candidate.

At Microsoft, we always offered drinks to our candidates, and one candidate "Jeff" took a pepsi. We got into my office, and he set it down on the desk. We started discussing his experiences and then launched into the whiteboard coding question, and he didn't get around to opening his pepsi.

We stood at the whiteboard, and Jeff started to write a line of code. He stopped to think about the overall algorithm, and absentmindedly took a step back in order to see the entire whiteboard. In doing so, he inadvertently knocked against the desk, and the pepsi fell off the edge.

This pepsi was still unopened. Thus, when it hit the ground, it *exploded on impact*.

Pepsi sprayed in foamy gusts in all directions from the can. It was a slow-motion moment as beige spots of soda splashed onto my white walls, my bookshelf, my keyboard. We both stood there frozen, our hands halfway out (too slow to catch the pepsi), looking at the dripping liquid coating the entire inside of my office.

We took a 5-minute break to get paper towels and mop up the mess. (Though my books always stuck together after that day, and my walls were never the same again.)

We then returned to the whiteboard question. Jeff was nervous by this time (understandably). He wrote some code, erased it, wrote more. He erased using his fingers against the board instead of using the eraser. Then sweat formed on his forehead, and he wiped it off using the same hand. By the end of the interview, his face was covered in streaks of red, green, and blue whiteboard marker.

I said, "I think you have some marker on your hands. I'll show you the restroom." and let the bathroom mirror show him the problem.

NOTES

NOTES

NOTES

NOTES