

Interview Questions - Microsoft, etc.

Disclaimer: This page is in no way connected with Microsoft Corporation, its affiliates, friends or haters

This page is moving to <http://www.softwareinterview.com>. Please update your bookmarks.

What?

Here is a set of questions that I have with me which software guys have asked at interviews in the past, most of them are actually from Microsoft but a few have been pulled together from other places too. I have collected these from friends and would welcome any additions from you. Mail them to kiran AT usc DOT edu. Do send me your solutions, but the intent of this page is to kindle enough interest in you to try similar logic and programming questions. PLEASE DO NOT REQUEST ME FOR MORE SOLUTIONS!

Cracking the Technical Interview: Third Edition (314 page e-book / PDF)

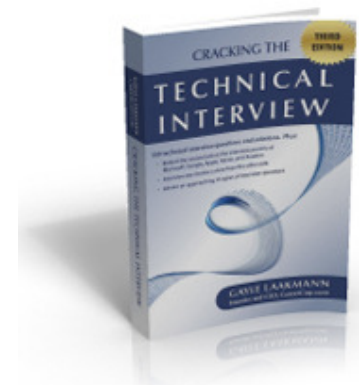
Delivered instantly as a PDF via email. *For Software Engineers & SDETs*

- **150 technical interview questions and solutions**, prepared by technical interview experts!
- **Advice** on how to approach 24 types of questions.
- **Behind the scenes** look at the Microsoft, Amazon, Google, Yahoo and Apple interview.
- **Interview war stories**: the View from the Other Side.

"Great book! It gave me the coding tips I needed to conquer the technical questions." - Deb (Accepted at Google)

30 Day Money Back Guarantee: Don't love the book? We'll give you your money back!

[Book Preview and More Info](#) | [CareerCup.com](#)



Just \$29 to Crack the Interview!



Why?

I have grown up reading Martin Gardner's Scientific American columns on Mathematical Games and interesting mathematics olympiad problems in high school. I discovered, to my delight, Bentley's *Programming Pearls* and David Gries' *The Science of Programming* in my computer science education. There is underlying beauty in mathematics and computer science. Some find it and others hate the subjects. When I found some of the Microsoft interview questions in graduate school, they were similar to mathematical puzzles that I was interested in. I started collecting them more as an illustration of interesting logic puzzles and algorithms in programming, than as an interview aid. If it has morphed into an interview questions page, that is more due to interest from all interviewees out there.

Some of the programming questions have a basic foundation in mathematics and algorithms. If the given data structure has a specific amount of information and the question asks you to extract/modify the given information, it is possible (though not always obvious) to find optimal solutions, and prove that it cannot be done any better, by quantifying the information content. But, there are always

elegant and ugly ways to extract the same information. When you manage to find out the optimal solution to a problem, it usually not only turns out to be elegant, but also has the "aha!" factor to it. Try proving on paper the following question from my undergraduate mid-term:

1. *Everyone Loves All Lovers*
 2. *Romeo Loves Juliet*
- Therefore, prove (i.e. 1 AND 2 \Rightarrow 3)
3. *I Love You*

How?

To answer one of the more frequent questions that I get: No! I have never interviewed at Microsoft. In fact, I work in a microprocessor company far removed from any software development (though I work mostly in programming). However, I have had two of my roommates and two more housemates eventually working in Microsoft, in addition to a large number of friends. None of them have contributed to this page AFTER they started working at Microsoft. Microsoft itself does not hold any patents and nor has it invented most of the questions. Most of them come from mathematics and physics books, party puzzles, programming lore and standard textbooks. I know many companies ask similar questions, though Microsoft might have made the practice more common (especially with the manhole cover type of questions).

Usage

I have spent some time in collecting with the intention of their value as a practice session. They are not meant to be exact questions that you need to know and answer in an interview. They are supposed to make you think! Discuss with your friends, colleagues, professors for answers (get your tuition money's worth). I have left the page in simple text format so you can print them out and try them on your flight to the interview in case you are pressed for time. One of the strangest phone calls I remember is from a girl, staying up in a Microsoft provided hotel room in Redmond the night before the interview, asking me about the solution for one of the questions from this page. I hope the rest of you are saner (or do not have my phone number).

[Kiran Bondalapati](#) **Ze Meat**

- [Puzzles, Riddles, etc.](#)
- [Programming Questions](#)
- [Computer Networks, Databases, etc.](#)
- [Computer Architecture](#)

Puzzles, Riddles and Others

0. Classic: If a bear walks one mile south, turns left and walks one mile to the east and then turns left again and walks one mile north and arrives at its original position, what is the color of the bear.

ANS. The color of the bear is trivial. The possible solutions to it are interesting.

In addition to the trivial north pole, there are additional circles near south pole. Think it out.

* 1. Given a rectangular (cuboidal for the puritans) cake with a rectangular piece removed (any size or orientation), how would you cut the remainder of the cake into two equal halves with one straight cut of a knife?

ANS. Join the centers of the original and the removed rectangle. It works for cuboids too! BTW, I have been getting many questions asking why a horizontal slice across the middle will not do. Please note the "any size or orientation" in the question! Don't get boxed in by the way you cut your birthday cake :) Think out of the box.

2. There are 3 baskets. one of them have apples, one has oranges only and the other has mixture of apples and oranges. The labels on their baskets always lie. (i.e. if the label says oranges, you are sure that it doesn't have oranges only, it could be a mixture) The task is to pick one basket and pick only one fruit from it and then correctly label all the three baskets.

HINT. There are only two combinations of distributions in which ALL the baskets have wrong labels. By picking a fruit from the one labeled MIXTURE, it is possible to tell what the other two baskets have.

3. You have 8 balls. One of them is defective and weighs less than others. You have a balance to measure balls against each other. In 2 weighings how do you find the defective one?

4. Why is a manhole cover round?

HINT. The diagonal of a square hole is larger than the side of a cover!

Alternate answers: 1. Round covers can be transported by one person, because they can be rolled on their edge. 2. A round cover doesn't need to be rotated to fit over a hole.

5. How many cars are there in the USA?

6. You've got someone working for you for seven days and a gold bar to pay them. The gold bar is segmented into seven connected pieces. You must give them a piece of gold at the end of every day. If you are only allowed to make two breaks in the gold bar, how do you pay your worker?

7. One train leaves Los Angeles at 15mph heading for New York. Another train leaves from New York at 20mph heading for Los Angeles on the same track. If a bird, flying at 25mph, leaves from Los Angeles at the same time as the train and flies back and forth between the two trains until they collide, how far will the bird have traveled?

HINT. Think relative speed of the trains.

8. You have two jars, 50 red marbles and 50 blue marbles. A jar will be picked at random, and then a marble will be picked from the jar. Placing all of the marbles

in the jars, how can you maximize the chances of a red marble being picked? What are the exact odds of getting a red marble using your scheme?

9. Imagine you are standing in front of a mirror, facing it. Raise your left hand. Raise your right hand. Look at your reflection. When you raise your left hand your reflection raises what appears to be his right hand. But when you tilt your head up, your reflection does too, and does not appear to tilt his/her head down. Why is it that the mirror appears to reverse left and right, but not up and down?

10. You have 5 jars of pills. Each pill weighs 10 gram, except for contaminated pills contained in one jar, where each pill weighs 9 gm. Given a scale, how could you tell which jar had the contaminated pills in just one measurement?

ANS. 1. Mark the jars with numbers 1, 2, 3, 4, and 5.

2. Take 1 pill from jar 1, take 2 pills from jar 2, take 3 pills from jar 3, take 4 pills from jar 4 and take 5 pills from jar 5.

3. Put all of them on the scale at once and take the measurement.

4. Now, subtract the measurement from 150 ($1*10 + 2*10 + 3*10 + 4*10 + 5*10$)

5. The result will give you the jar number which has contaminated pill.

11. If you had an infinite supply of water and a 5 quart and 3 quart pail, how would you measure exactly 4 quarts?

12. You have a bucket of jelly beans. Some are red, some are blue, and some green. With your eyes closed, pick out 2 of a like color. How many do you have to grab to be sure you have 2 of the same?

13. Which way should the key turn in a car door to unlock it?

14. If you could remove any of the 50 states, which state would it be and why?

15. There are four dogs/ants/people at four corners of a square of unit distance. At the same instant all of them start running with unit speed towards the person on their clockwise direction and will always run towards that target. How long does it take for them to meet and where?

HINT. They will meet in the center and the distance covered by them is independent of the path they actually take (a spiral).

16. (from Tara Hovel) A helicopter drops two trains, each on a parachute, onto a straight infinite railway line. There is an undefined distance between the two trains. Each faces the same direction, and upon landing, the parachute attached to each train falls to the ground next to the train and detaches. Each train has a microchip that controls its motion. The chips are identical. There is no way for the trains to know where they are. You need to write the code in the chip to make the trains bump into each other. Each line of code takes a single clock cycle to execute.

You can use the following commands (and only these);

MF - moves the train forward

MB - moves the train backward

IF (P) - conditional that's satisfied if the train is next to a parachute. There is no

"then" to this IF statement.
GOTO

ANS.

A: MF

IF (P)

GOTO B

GOTO A

B: MF

GOTO B

Explanation: The first line simply gets them off the parachutes. You need to get the trains off their parachutes so the back train can find the front train's parachute, creating a special condition that will allow it to break out of the code they both have to follow initially. They both loop through A: until the back train finds the front train's parachute, at which point it goes to B: and gets stuck in that loop. The front train still hasn't found a parachute, so it keeps in the A loop. Because each line of code takes a "clock cycle" to execute, it takes longer to execute the A loop than the B loop, therefore the back train (running in the B loop) will catch up to the front train.

Personality

It is best to read some website or a book for questions like these.

1. Tell me the courses you liked and why did you like them.
2. Give an instance in your life in which you were faced with a problem and you tackled it successfully.
3. What is your ideal working environment.
4. Why do you think you are smart.
5. Questions on the projects listed on the Resume.
6. Do you want to know any thing about the company.(Try to ask some relevant and interesting question).
7. How long do you want to stay in USA and why (I guess non-citizens get this)?
8. What is your geographical preference?
9. What are your expectations from the job.

Algorithms and Programming

1. Given a rectangular (cuboidal for the puritans) cake with a rectangular piece removed (any size or orientation), how would you cut the remainder of the cake into two equal halves with one straight cut of a knife ?

2. You're given an array containing both positive and negative integers and required to find the sub-array with the largest sum ($O(N)$ a la KBL). Write a routine in C for the above.
3. Given an array of size N in which every number is between 1 and N , determine if there are any duplicates in it. You are allowed to destroy the array if you like. [I ended up giving about 4 or 5 different solutions for this, each supposedly better than the others].
4. Write a routine to draw a circle ($x^2 + y^2 = r^2$) without making use of any floating point computations at all. [This one had me stuck for quite some time and I first gave a solution that did have floating point computations].
5. Given only putchar (no sprintf, itoa, etc.) write a routine putlong that prints out an unsigned long in decimal. [I gave the obvious solution of taking $\% 10$ and $/ 10$, which gives us the decimal value in reverse order. This requires an array since we need to print it out in the correct order. The interviewer wasn't too pleased and asked me to give a solution which didn't need the array].
6. Give a one-line C expression to test whether a number is a power of 2. [No loops allowed - it's a simple test.]
7. Given an array of characters which form a sentence of words, give an efficient algorithm to reverse the order of the words (not characters) in it.
8. How many points are there on the globe where by walking one mile south, one mile east and one mile north you reach the place where you started.
9. Give a very good method to count the number of ones in a "n" (e.g. 32) bit number.

ANS. Given below are simple solutions, find a solution that does it in $\log(n)$ steps.

Iterative

```
function iterativecount (unsigned int n)
begin
    int count=0;
    while (n)
    begin
        count += n & 0x1 ;
        n >>= 1;
    end
    return count;
end
```

Sparse Count

```
function sparsecount (unsigned int n)
begin
    int count=0;
    while (n)
    begin
```

```
count++;  
n &= (n-1);  
end  
return count ;  
end
```

10. What are the different ways to implement a condition where the value of x can be either a 0 or a 1. Apparently the if then else solution has a jump when written out in assembly. if (x == 0) y=a else y=b There is a logical, arithmetic and a data structure solution to the above problem.

11. Reverse a linked list.

12. Insert in a sorted list

13. In a X's and O's game (i.e. TIC TAC TOE) if you write a program for this give a fast way to generate the moves by the computer. I mean this should be the fastest way possible.

The answer is that you need to store all possible configurations of the board and the move that is associated with that. Then it boils down to just accessing the right element and getting the corresponding move for it. Do some analysis and do some more optimization in storage since otherwise it becomes infeasible to get the required storage in a DOS machine.

14. I was given two lines of assembly code which found the absolute value of a number stored in two's complement form. I had to recognize what the code was doing. Pretty simple if you know some assembly and some fundaes on number representation.

15. Give a fast way to multiply a number by 7.

16. How would go about finding out where to find a book in a library. (You don't know how exactly the books are organized beforehand).

17. Linked list manipulation.

18. Tradeoff between time spent in testing a product and getting into the market first.

19. What to test for given that there isn't enough time to test everything you want to.

20. First some definitions for this problem: a) An ASCII character is one byte long and the most significant bit in the byte is always '0'. b) A Kanji character is two bytes long. The only characteristic of a Kanji character is that in its first byte the most significant bit is '1'.

Now you are given an array of a characters (both ASCII and Kanji) and, an index into the array. The index points to the start of some character. Now you need to write a function to do a backspace (i.e. delete the character before the given index).

21. Delete an element from a doubly linked list.
22. Write a function to find the depth of a binary tree.
23. Given two strings S1 and S2. Delete from S2 all those characters which occur in S1 also and finally create a clean S2 with the relevant characters deleted.
24. Assuming that locks are the only reason due to which deadlocks can occur in a system. What would be a foolproof method of avoiding deadlocks in the system.
25. Reverse a linked list.

Ans: Possible answers -

iterative loop

```
curr->next = prev;  
prev = curr;  
curr = next;  
next = curr->next  
endloop
```

recursive reverse(ptr)

```
if (ptr->next == NULL)  
return ptr;  
temp = reverse(ptr->next);  
temp->next = ptr;  
return ptr;  
end
```

26. Write a small lexical analyzer - interviewer gave tokens. expressions like "a*b" etc.

27. Besides communication cost, what is the other source of inefficiency in RPC? (answer : context switches, excessive buffer copying). How can you optimize the communication? (ans : communicate through shared memory on same machine, bypassing the kernel _ A Univ. of Wash. thesis)

28. Write a routine that prints out a 2-D array in spiral order!

29. How is the readers-writers problem solved? - using semaphores/ada .. etc.

30. Ways of optimizing symbol table storage in compilers.

31. A walk-through through the symbol table functions, lookup() implementation etc. - The interviewer was on the Microsoft C team.

32. A version of the "There are three persons X Y Z, one of which always lies".. etc..

33. There are 3 ants at 3 corners of a triangle, they randomly start moving towards another corner.. what is the probability that they don't collide.

34. Write an efficient algorithm and C code to shuffle a pack of cards.. this one was a feedback process until we came up with one with no extra storage.
35. The if (x == 0) y = 0 etc..
36. Some more bitwise optimization at assembly level
37. Some general questions on Lex, Yacc etc.
38. Given an array t[100] which contains numbers between 1..99. Return the duplicated value. Try both O(n) and O(n-square).
39. Given an array of characters. How would you reverse it. ? How would you reverse it without using indexing in the array.
40. Given a sequence of characters. How will you convert the lower case characters to upper case characters. (Try using bit vector - solutions given in the C lib -typec.h)
41. Fundamentals of RPC.
42. Given a linked list which is sorted. How will u insert in sorted way.
43. Given a linked list How will you reverse it.
44. Give a good data structure for having n queues (n not fixed) in a finite memory segment. You can have some data-structure separate for each queue. Try to use at least 90% of the memory space.
45. Do a breadth first traversal of a tree.
46. Write code for reversing a linked list.
47. Write, efficient code for extracting unique elements from a sorted list of array. e.g. (1, 1, 3, 3, 3, 5, 5, 5, 9, 9, 9, 9) -> (1, 3, 5, 9).
48. Given an array of integers, find the contiguous sub-array with the largest sum.

ANS. Can be done in O(n) time and O(1) extra space. Scan array from 1 to n. Remember the best sub-array seen so far and the best sub-array ending in i.

49. Given an array of length N containing integers between 1 and N, determine if it contains any duplicates.

ANS. [Is there an O(n) time solution that uses only O(1) extra space and does not destroy the original array?]

50. Sort an array of size n containing integers between 1 and K, given a temporary scratch integer array of size K.

ANS. Compute cumulative counts of integers in the auxiliary array. Now scan the

original array, rotating cycles! [Can someone word this more nicely?]

* 51. An array of size k contains integers between 1 and n . You are given an additional scratch array of size n . Compress the original array by removing duplicates in it. What if $k \ll n$?

ANS. Can be done in $O(k)$ time i.e. without initializing the auxiliary array!

52. An array of integers. The sum of the array is known not to overflow an integer. Compute the sum. What if we know that integers are in 2's complement form?

ANS. If numbers are in 2's complement, an ordinary looking loop like `for(i=total=0; i < n; total+=array[i++]);` will do. No need to check for overflows!

53. An array of characters. Reverse the order of words in it.

ANS. Write a routine to reverse a character array. Now call it for the given array and for each word in it.

* 54. An array of integers of size n . Generate a random permutation of the array, given a function `rand_n()` that returns an integer between 1 and n , both inclusive, with equal probability. What is the expected time of your algorithm?

ANS. "Expected time" should ring a bell. To compute a random permutation, use the standard algorithm of scanning array from n down to 1, swapping i -th element with a uniformly random element $\leq i$ -th. To compute a uniformly random integer between 1 and k ($k < n$), call `rand_n()` repeatedly until it returns a value in the desired range.

55. An array of pointers to (very long) strings. Find pointers to the (lexicographically) smallest and largest strings.

ANS. Scan array in pairs. Remember largest-so-far and smallest-so-far. Compare the larger of the two strings in the current pair with largest-so-far to update it. And the smaller of the current pair with the smallest-so-far to update it. For a total of $\leq 3n/2$ `strcmp()` calls. That's also the lower bound.

56. Write a program to remove duplicates from a sorted array.

```
ANS. int remove_duplicates(int * p, int size)
{
    int current, insert = 1;
    for (current=1; current < size; current++)
        if (p[current] != p[insert-1])
        {
            p[insert] = p[current];
            current++;
            insert++;
        } else
            current++;
}
```

```
return insert;
```

```
}
```

57. C++ (what is virtual function ? what happens if an error occurs in constructor or destructor. Discussion on error handling, templates, unique features of C++. What is different in C++, (compare with unix).

58. Given a list of numbers (fixed list) Now given any other list, how can you efficiently find out if there is any element in the second list that is an element of the first list (fixed list).

59. Given 3 lines of assembly code : find it is doing. IT was to find absolute value.

60. If you are on a boat and you throw out a suitcase, Will the level of water increase.

61. Print an integer using only putchar. Try doing it without using extra storage.

62. Write C code for (a) deleting an element from a linked list (b) traversing a linked list

63. What are various problems unique to distributed databases

64. Declare a void pointer ANS. void *ptr;

65. Make the pointer aligned to a 4 byte boundary in a efficient manner ANS. Assign the pointer to a long number and the number with 11...1100 add 4 to the number

66. What is a far pointer (in DOS)

67. What is a balanced tree

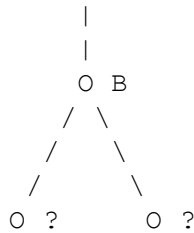
68. Given a linked list with the following property node2 is left child of node1, if node2 < node1 else, it is the right child.

```

O P
|
|
O A
|
|
O B
|
|
O C

```

How do you convert the above linked list to the form without disturbing the property. Write C code for that.



determine where do A and C go

69. Describe the file system layout in the UNIX OS

ANS. describe boot block, super block, inodes and data layout

70. In UNIX, are the files allocated contiguous blocks of data

ANS. no, they might be fragmented

How is the fragmented data kept track of

ANS. Describe the direct blocks and indirect blocks in UNIX file system

71. Write an efficient C code for 'tr' program. 'tr' has two command line arguments. They both are strings of same length. tr reads an input file, replaces each character in the first string with the corresponding character in the second string. eg. 'tr abc xyz' replaces all 'a's by 'x's, 'b's by 'y's and so on. ANS.

a) have an array of length 26.

put 'x' in array element corr to 'a'

put 'y' in array element corr to 'b'

put 'z' in array element corr to 'c'

put 'd' in array element corr to 'd'

put 'e' in array element corr to 'e'

and so on.

the code

```
while (!eof)
```

```
{
```

```
c = getc();
```

```
putc(array[c - 'a']);
```

```
}
```

72. what is disk interleaving

73. why is disk interleaving adopted

74. given a new disk, how do you determine which interleaving is the best a) give 1000 read operations with each kind of interleaving determine the best interleaving from the statistics

75. draw the graph with performance on one axis and 'n' on another, where 'n' in the 'n' in n-way disk interleaving. (a tricky question, should be answered carefully)

76. I was a c++ code and was asked to find out the bug in that. The bug was that he declared an object locally in a function and tried to return the pointer to that object. Since the object is local to the function, it no more exists after returning from the function. The pointer, therefore, is invalid outside.

77. A real life problem - A square picture is cut into 16 squares and they are shuffled. Write a program to rearrange the 16 squares to get the original big square.

```
78.
int *a;
char *c;
*(a) = 20;
*c = *a;
printf("%c",*c);
```

what is the output?

79. Write a program to find whether a given m/c is big-endian or little-endian!

80. What is a volatile variable?

81. What is the scope of a static function in C ?

82. What is the difference between "malloc" and "calloc"?

```
83. struct n { int data; struct n* next}node;
node *c,*t;
c->data = 10;
t->next = null;
*c = *t;
```

what is the effect of the last statement?

84. If you're familiar with the ? operator $x ? y : z$ you want to implement that in a function: `int cond(int x, int y, int z);` using only `~, !, ^, &, +, |, <<, >>` no if statements, or loops or anything else, just those operators, and the function should correctly return y or z based on the value of x. You may use constants, but only 8 bit constants. You can cast all you want. You're not supposed to use extra variables, but in the end, it won't really matter, using vars just makes things cleaner. You should be able to reduce your solution to a single line in the end though that requires no extra vars.

85. You have an abstract computer, so just forget everything you know about computers, this one only does what I'm about to tell you it does. You can use as many variables as you need, there are no negative numbers, all numbers are integers. You do not know the size of the integers, they could be infinitely large, so you can't count on truncating at any point. There are NO comparisons allowed, no if statements or anything like that. There are only four operations you can do on a variable.

- 1) You can set a variable to 0.
- 2) You can set a variable = another variable.
- 3) You can increment a variable (only by 1), and it's a post increment.

4) You can loop. So, if you were to say `loop(v1)` and `v1 = 10`, your loop would execute 10 times, but the value in `v1` wouldn't change so the first line in the loop can change value of `v1` without changing the number of times you loop.

You need to do 3 things.

1) Write a function that decrements by 1.

2) Write a function that subtracts one variable from another.

3) Write a function that divides one variable by another.

4) See if you can implement all 3 using at most 4 variables. Meaning, you're not making function calls now, you're making macros. And at most you can have 4 variables. The restriction really only applies to divide, the other 2 are easy to do with 4 vars or less. Division on the other hand is dependent on the other 2 functions, so, if subtract requires 3 variables, then divide only has 1 variable left unchanged after a call to subtract. Basically, just make your function calls to decrement and subtract so you pass your vars in by reference, and you can't declare any new variables in a function, what you pass in is all it gets.

Linked lists

* 86. Under what circumstances can one delete an element from a singly linked list in constant time?

ANS. If the list is circular and there are no references to the nodes in the list from anywhere else! Just copy the contents of the next node and delete the next node. If the list is not circular, we can delete any but the last node using this idea. In that case, mark the last node as dummy!

* 87. Given a singly linked list, determine whether it contains a loop or not.

ANS. (a) Start reversing the list. If you reach the head, gotcha! there is a loop! But this changes the list. So, reverse the list again.

(b) Maintain two pointers, initially pointing to the head. Advance one of them one node at a time. And the other one, two nodes at a time. If the latter overtakes the former at any time, there is a loop!

```
p1 = p2 = head;

do {
    p1 = p1->next;
    p2 = p2->next->next;
} while (p1 != p2);
```

88. Given a singly linked list, print out its contents in reverse order. Can you do it without using any extra space?

ANS. Start reversing the list. Do this again, printing the contents.

89. Given a binary tree with nodes, print out the values in pre-order/in-order/post-order without using any extra space.

90. Reverse a singly linked list recursively. The function prototype is `node *reverse (node *)` ;

ANS.

```

node * reverse (node * n)
{
    node * m ;

    if (! (n && n -> next))
        return n ;

    m = reverse (n -> next) ;
    n -> next -> next = n ;
    n -> next = NULL ;
    return m ;
}

```

91. Given a singly linked list, find the middle of the list.

HINT. Use the single and double pointer jumping. Maintain two pointers, initially pointing to the head. Advance one of them one node at a time. And the other one, two nodes at a time. When the double reaches the end, the single is in the middle. This is not asymptotically faster but seems to take less steps than going through the list twice.

Bit-manipulation

92. Reverse the bits of an unsigned integer.

ANS.

```

#define reverse(x) \
    (x=x>>16| (0x0000ffff&x)<<16, \
    x=(0xff00ff00&x)>>8| (0x00ff00ff&x)<<8, \
    x=(0xf0f0f0f0&x)>>4| (0x0f0f0f0f&x)<<4, \
    x=(0xcccccccc&x)>>2| (0x33333333&x)<<2, \
    x=(0xaaaaaaaa&x)>>1| (0x55555555&x)<<1)

```

* 93. Compute the number of ones in an unsigned integer.

ANS.

```

#define count_ones(x) \
    (x=(0xaaaaaaaa&x)>>1+(0x55555555&x), \
    x=(0xcccccccc&x)>>2+(0x33333333&x), \
    x=(0xf0f0f0f0&x)>>4+(0x0f0f0f0f&x), \
    x=(0xff00ff00&x)>>8+(0x00ff00ff&x), \
    x=x>>16+(0x0000ffff&x))

```

94. Compute the discrete log of an unsigned integer.

ANS.

```

#define discrete_log(h) \
    (h=(h>>1)|(h>>2), \
    h|=(h>>2), \
    h|=(h>>4), \

```

```

h|=(h>>8), \
h|=(h>>16), \
h=(0xaaaaaaaa&h)>>1+(0x55555555&h), \
h=(0xcccccccc&h)>>2+(0x33333333&h), \
h=(0xf0f0f0f0&h)>>4+(0x0f0f0f0f&h), \
h=(0xff00ff00&h)>>8+(0x00ff00ff&h), \
h=(h>>16)+(0x0000ffff&h)

```

If I understand it right, $\log_2(2) = 1$, $\log_2(3) = 1$, $\log_2(4) = 2$ But this macro does not work out $\log_2(0)$ which does not exist! How do you think it should be handled?

* 95. How do we test most simply if an unsigned integer is a power of two?

ANS. `#define power_of_two(x) \ ((x)&&(~(x&(x-1))))`

96. Set the highest significant bit of an unsigned integer to zero.

ANS. (from Denis Zabavchik) Set the highest significant bit of an unsigned integer to zero

```

#define zero_most_significant(h) \
(h&=(h>>1)|(h>>2), \
h|=(h>>2), \
h|=(h>>4), \
h|=(h>>8), \
h|=(h>>16))

```

97. Let $f(k) = y$ where k is the y -th number in the increasing sequence of non-negative integers with the same number of ones in its binary representation as y , e.g. $f(0) = 1$, $f(1) = 1$, $f(2) = 2$, $f(3) = 1$, $f(4) = 3$, $f(5) = 2$, $f(6) = 3$ and so on. Given $k \geq 0$, compute $f(k)$.

Others

98. A character set has 1 and 2 byte characters. One byte characters have 0 as the first bit. You just keep accumulating the characters in a buffer. Suppose at some point the user types a backspace, how can you remove the character efficiently. (Note: You cant store the last character typed because the user can type in arbitrarily many backspaces)

99. What is the simplest way to check if the sum of two unsigned integers has resulted in an overflow.

100. How do you represent an n -ary tree? Write a program to print the nodes of such a tree in breadth first order.

101. Write the 'tr' program of UNIX. Invoked as

`tr -str1 -str2`. It reads stdin and prints it out to stdout, replacing every occurrence of `str1[i]` with `str2[i]`.

e.g. `tr -abc -xyz`

to be and not to be <- input
to ye xnd not to ye <- output

Networks and Security

1. How do you use RSA for both authentication and secrecy?
2. What is ARP and how does it work?
3. What's the difference between a switch and a router?
4. Name some routing protocols? (RIP, OSPF etc..)
5. How do you do authentication with message digest(MD5)? (Usually MD is used for finding tampering of data)
6. How do you implement a packet filter that distinguishes following cases and selects first case and rejects second case.
 - i) A host inside the corporate n/w makes a ftp request to outside host and the outside host sends reply.
 - ii) A host outside the network sends a ftp request to host inside. for the packet filter in both cases the source and destination fields will look the same.
7. How does traceroute work? Now how does traceroute make sure that the packet follows the same path that a previous (with ttl - 1) probe packet went in?
8. Explain Kerberos Protocol ?
9. What are digital signatures and smart cards?
10. Difference between discretionary access control and mandatory access control?

Java

1. How do you find the size of a java object (not the primitive type) ?
ANS. type cast it to string and find its s.length()
2. Why is multiple inheritance not provided in Java?
3. Thread t = new Thread(); t.start(); t = null; now what will happen to the created thread?
4. How is garbage collection done in java?
5. How do you write a "ping" routine in java?

6. What are the security restrictions on applets?

Graphics

1. Write a function to check if two rectangles defined as below overlap or not.
`struct rect { int top, bot, left, right; } r1, r2;`
2. Write a `SetPixel(x, y)` function, given a pointer to the bitmap. Each pixel is represented by 1 bit. There are 640 pixels per row. In each byte, while the bits are numbered right to left, pixels are numbered left to right. Avoid multiplications and divisions to improve performance.

Databases

- * 1. You, a designer want to measure disk traffic i.e. get a histogram showing the relative frequency of I/O/second for each disk block. The buffer pool has `b` buffers and uses LRU replacement policy. The disk block size and buffer pool block sizes are the same. You are given a routine `int lru_block_in_position (int i)` which returns the `block_id` of the block in the `i`-th position in the list of blocks managed by LRU. Assume position 0 is the hottest. You can repeatedly call this routine. How would you get the histogram you desire?

Hints and Answers

1. Simply do histogram `[lru_block_in_position (b-1)] ++` at frequent intervals... The sampling frequency should be close to the disk I/O rate. It can be adjusted by remembering the last block seen in position `b`. If same, decrease frequency; if different, increase, with exponential decay etc. And of course, take care of overflows in the histogram.

Semaphores

1. Implement a multiple-reader-single-writer lock given a compare-and-swap instruction. Readers cannot overtake waiting writers.
-

Computer Architecture

1. Explain what is DMA?
2. What is pipelining?
3. What are superscalar machines and vliw machines?
4. What is cache?
5. What is cache coherency and how is it eliminated?
6. What is write back and write through caches?
7. What are different pipelining hazards and how are they eliminated.
8. What are different stages of a pipe?
9. Explain more about branch prediction in controlling the control hazards
10. Give examples of data hazards with pseudo codes.
11. How do you calculate the number of sets given its way and size in a cache?

12. How is a block found in a cache?
 13. Scoreboard analysis.
 14. What is miss penalty and give your own ideas to eliminate it.
 15. How do you improve the cache performance.
 16. Different addressing modes.
 17. Computer arithmetic with two's complements.
 18. About hardware and software interrupts.
 19. What is bus contention and how do you eliminate it.
 20. What is aliasing?
 - 21) What is the difference between a latch and a flip flop?
 - 22) What is the race around condition? How can it be overcome?
 - 23) What is the purpose of cache? How is it used?
 - 24) What are the types of memory management?
-

[Kiran Bondalapati Homepage](http://halcyon.usc.edu/~kiran/msqs.html)