1. 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)

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

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

4. Ways of optimizing symbol table storage in compilers.

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

6. 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 << n?

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

7. 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!

8. 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.

9. 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 downto 1, swapping i-th element with a uniformly random element <= 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.

10. 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 <= 3n/2 strcmp() calls. That's also the lower bound.

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

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

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

14. What are various problems unique to distributed databases

15. Declare a void pointer

ANS. void *ptr;

16. 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))
```

17. 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 >= 0, compute f(k).

18. 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)

19. 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)
```

20. 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))
```