http://www.techinterview.org/Puzzles/fog0000000080.html

reverse a string - word by word aha:!!

a typical programming interview question is "reverse a string, in place". if you understand pointers, the solution is simple. even if you don't, it can be accomplished using array indices. i usually ask candidates this question first, so they get the algorithm in their head. then i play dirty by asking them to reverse the string word by word, in place. for example if our string is "the house is blue", the return value would be "blue is house the". the words are reversed, but the letters are still in order (within the word).

solution: solving the initial problem of just reversing a string can either be a huge help or a frustrating hinderance. most likely the first attempt will be to solve it the same way, by swapping letters at the front of the string with letters at the back, and then adding some logic to keep the words in order. this attempt will lead to confusion pretty quickly.

for example, if we start by figuring out that "the" is 3 letters long and then try to put the "t" from "the" where the "l" from "blue" is, we encounter a problem. where do we put the "l" from "blue"? hmm... well we could have also figured out how long "blue" was and that would tell us where to put the "l" at... but the "e" from "blue" needs to go into the space after "the". argh. its getting quite confusing. in fact, i would be delighted to even see a solution to this problem using this attack method. i don't think its impossible, but i think it is so complex that it's not worth pursuing.

here's a hint. remember before when we just reversed "the house is blue"? what happened?

```
initial: the house is blue
reverse: eulb si esuoh eht
```

look at the result for a minute. notice anything? if you still don't see it, try this.

```
initial: the house is blue
reverse: eulb si esuoh eht
wanted : blue is house the
```

the solution can be attained by first reversing the string normally, and then just reversing each word

problem: you have 100 doors in a row that are all initially closed. you make 100 passes by the doors starting with the first door every time. the first time through you visit every door and toggle the door (if the door is closed, you open it, if its open, you close it). the second time you only visit every 2nd door (door #2, #4, #6). the third time, every 3rd door (door #3, #6, #9), etc, until you only visit the 100th door.

question: what state are the doors in after the last pass? which are open which are closed?

problem: you have 100 doors in a row that are all initially closed. you make 100 passes by the doors starting with the first door every time. the first time through you visit every door and toggle the door (if the door is closed, you open it, if its open, you close it). the second time you only visit every 2nd door (door #2, #4, #6). the third time, every 3rd door (door #3, #6, #9), etc, until you only visit the 100th door.

for example, after the first pass every door is open. on the second pass you only visit the even doors (2,4,6,8...) so now the even doors are closed and the odd ones are opened. the third time through you will close door 3 (opened from the first pass), open door 6 (closed from the second pass), etc..

question: what state are the doors in after the last pass? which are open which are closed?

solution: you can figure out that for any given door, say door #42, you will visit it for every divisor it has. so 42 has 1 & 42, 2 & 21, 3 & 14, 6 & 7. so on pass 1 i will open the door, pass 2 i will close it, pass 3 open, pass 6 close, pass 7 open, pass 14 close, pass 21 open, pass 42 close. for every pair of divisors the door will just end up back in its initial state. so you might think that every door will end up closed? well what about door #9. 9 has the divisors 1 & 9, 3 & 3. but 3 is repeated because 9 is a perfect square, so you will only visit door #9, on pass 1, 3, and 9... leaving it open at the end. only perfect square doors will be open at the end.

---

problem: you have two jars, 50 red marbles, 50 blue marbles. you need to place all the marbles into the jars such that when you blindly pick one marble out of one jar, you maximize the chances that it will be red. (when picking, you'll first randomly pick a jar, and then randomly pick a marble out of that jar) you can arrange the marbles however you like, but each marble must be in a jar.

solution: chance! chance is easy if you know how to do the formula. we know that we have two choices to make. first we'll pick a jar, and each jar will have a 1/2 chance of being picked. then we'll pick a marble, and depending how we stack the marbles, we'll have a (# of red marbles in jar)/(# of total marbles in jar) chance of getting a red one.

for example, say we put all the red marbles into **jar A** and all the blue ones into **jar B**. then our chances for picking a red one are:

1/2 chance we pick **jar A** * 50/50 chance we pick a red marble
1/2 chance we pick **jar B** * 0/50 chance we pick a red marble

do the math and you get 1/2 chance for a red marble from **jar A** and a 0/2 chance for a red marble from **jar B**. add 'em up and you get the result = 1/2 chance for picking a red marble.

think about it for awhile and see if you can figure out the right combination. we had a 50/50 (guaranteed) chance in picking a red marble from **jar A**, but we didn't have to have 50 red marbles in there to guarantee those fantastic odds, did we? we could've just left 1 red marble in there and the odds are still 1/1. then we can take all those other marbles and throw them in **jar B** to help the odds out there.

let's look at those chances:

1/2 we pick **jar A** * 1/1 we pick a red marble
1/2 we pick **jar B** * 49/99 we pick a red marble

do the math and add them up to get 1/2 + 49/198 = 148/198, which is almost 3/4.

we can prove these are the best odds in a somewhat non-formal way as follows. our goal is to maximize the odds of picking a red marble. therefore we can subdivide this goal into maximizing the odds of picking a red marble in **jar A** and maximizing the odds of picking a red marble in **jar B**. if we do that, then we will have achieved our goal. it is true that by placing more red marbles into a jar we will increase the chances of picking a red marble. it is also true that by reducing the number of blue marbles in a jar we will increase the odds also. we've maximized the odds in **jar A** since 1/1 is the maximum odds by reducing the number of blue marbles to 0 (the minimum). we've also maximized the number of red marbles in **jar B**. if we added any more red marbles to **jar B** we would have to take them out of **jar A** which reduce the odds there to 0 (very bad). if we took any more blue ones out of **jar B** we would have to put them in **jar A** which reduce the odds there by 50% (very bad).

---

problem: two trains enter a tunnel 200 miles long (yeah, its a big tunnel) travelling at 100 mph at the same time from opposite directions. as soon as they enter the tunnel a supersonic bee flying at 1000 mph starts from one train and heads toward the other one. as soon as it reaches the other one it turns around and heads back toward the first, going back and forth between the trains until the trains collide in a fiery explosion in the middle of the tunnel (the bee survives). how far did the bee travel?

solution: this puzzle falls pretty high on my aha scale. my first inclination when i heard it was to think "ok, so i just need to sum up the distances that the bee travels..." but then you quickly realize that its a difficult (not impossible) summation which the interviewer could hardly expect you to answer (unless i guess if you are looking for a job as a quant). "there must be a trick" you say. eh, sort of i guess, enough to say that this question is a stupid interview question.

the tunnel is 200 miles long. the trains meet in the middle travelling at 100 mph, so it takes them an hour to reach the middle. the bee is travelling 1000 mph for an hour (since its flying the whole time the trains are racing toward one another) - so basically the bee goes 1000 miles.

there is no process to explain, so this question can't possibly teach you anything about the person. they either know it or they don't and if they already knew it before you asked, you're not going to be able to tell when they give you the answer. so don't ask this question. and if someone asks you this question, just tell them you've already heard it before.

---

problem: write the definition for this function *without using any built-in functions*. if pStr is null, return 0. if pStr contains non-numeric characters, either return 0 (ok) or return the number derived so far (better) (e.g. if its "123A", then return 123). assume all numbers are positive. plus or minus signs can be considered non-numeric characters. in order to solve this program, the programmer must understand the difference between the integer 0 and the character '0', and how converting '0' to an int, will not result in 0. in other words, they have to understand what ascii is all about.

string manipulation functions are great programming questions. they test whether the user can understand and translate into code simple algorithms. string functions test pointer arithmetic which usually shows a knowledgeable programmer. also there are usually multiple solutions, some more efficient than others. plus people use them all the time so they should understand how they work. my favorite is atoi and i start the problem like this:

```
int atoi( char* pStr )
```

write the definition for this function *without using any built-in functions*. if pStr is null, return 0. if pStr contains non-numeric characters, either return 0 (ok) or return the number derived so far (better) (e.g. if its "123A", then return 123). assume all numbers are positive. plus or minus signs can be considered non-numeric characters. in order to solve this program, the programmer must understand the difference between the integer 0 and the character '0', and how converting '0' to an int, will not result in 0. in other words, they have to understand what ascii is all about. if they are stuck solving this problem, just ask them first to write:

```
charToInt(char c)
```

if they can't do that then they basically missed half the problem. any moderately talented programmer who has a CS degree knows how to convert a char to an int. (note i said convert, not cast. `charToInt('9')` should return 9.)

when they start to solve the problem you will notice that they must make a choice in how they will process the string - from left to right or right to left. i will discuss both methods and the difficulties encountered in each.

"right to left" - this method starts at the right hand letter of the string and converts that character to an int. it then stores this value after promoting it to its correct "tens" place.

```
int atoi( char* pStr )
{
  int iRetVal = 0;
  int iTens = 1;

  if ( pStr )
  {
    char* pCur = pStr;
    while (*pCur)
      pCur++;

    pCur--;

    while ( pCur >= pStr && *pCur <= '9' && *pCur >= '0' )
    {
      iRetVal += ((*pCur - '0') * iTens);
      pCur--;
      iTens *= 10;
    }
  }
  return iRetVal;
}
```

"left to right" - this method keeps adding the number and multiplying the result by ten before continuing to the next number. e.g. if you had "6234" and you processed from left to right you'd have 6, then if you kept reading you'd multiply your result by 10 (6*10) to add a zero for where the next number would go. 60, and then you'd slide the 2 into the zero place you just made. 62. do it again, 620, slide the next number in, 623.

```
int atoi( char* pStr )
{
  int iRetVal = 0;

  if ( pStr )
  {
    while ( *pStr && *pStr <= '9' && *pStr >= '0' )
    {
      iRetVal = (iRetVal * 10) + (*pStr - '0');
      pStr++;
    }
  }
  return iRetVal;
}
```

i think the "left to right" method is a little bit cleaner, or maybe its just cooler. but both are "correct".

remember that debugging code on paper is somewhat hard. most programmers aren't used to studying code that much when you can just hit F-7, compile and see if the compiler barfs or not. if you notice an error, just ask them to step through a sample string drawing out what is happening with all the variables and the pointers in every step. they should find their mistake then and fix it (no points deducted).

---

two MIT math grads bump into each other at Fairway on the upper west side. they haven't seen each other in over 20 years.

the first grad says to the second: "how have you been?"
second: "great! i got married and i have three daughters now"
first: "really? how old are they?"
second: "well, the product of their ages is 72, and the sum of their ages is the same as the number on that building over there.."
first: "right, ok.. oh wait.. hmm, i still don't know"
second: "oh sorry, the oldest one just started to play the piano"
first: "wonderful! my oldest is the same age!"

problem: how old are the daughters?

solution: start with what you know. you know there are 3 daughters whose ages multiply to 72. let's look at the possibilities...

Ages:	**Sum of ages:**
1 1 72	74
1 2 36	39
1 3 24	28
1 4 18	23
1 6 12	19
1 8 9	18
2 2 18	22
2 3 12	17
2 4 9	15
2 6 6	14
3 3 8	14
3 4 6	13

after looking at the building number the man still can't figure out what their ages are (we're assuming since he's an MIT math grad, he can factor 72 and add up the sums), so the building number must be 14, since that is the only sum that has more than one possibility.

finally the man discovers that there is an oldest daughter. that rules out the "2 6 6" possibility since the two oldest would be twins. therefore, the daughters ages must be "3 3 8".
(caveat: an astute reader pointed out that it **is** possible for two siblings to have the same age but not be twins, for instance one is born in january, and the next is conceived right away and delivered in october. next october both siblings will be one year old. if a candidate points this out, extra credit points to him/her.)

this question is pretty neat, although there is certainly a bit of an aha factor to it. the clues are given in such a way that you think you are missing information (the building number), but whats important isn't the building number, but the fact that the first man thought that it was enough information, but actually wasn't.

even if the candidate doesn't know the solution, they could come up with some interesting thoughts. if they just stare at you and shrug "i dunno" then thank them for their time and don't give them a fogcreek pen.

---

problem: this year on October 2, 2001, the date in MMDDYYYY format will be a palindrome (same forwards as backwards).
10/02/2001
when was the last date that this occurred on? (see if you can do it in your head!)

solution: we know the year has to be less than 2001 since we already have the palindrome for 10/02. it can't be any year in 1900 because that would result in a day of 91. same for 1800 down to 1400. it could be a year in 1300 because that would be the 31st day. so whats the latest year in 1300 that would make a month? at first i thought it would be 1321, since that would give us the 12th month, but we have to remember that we want the maximum **year** in the 1300 century with a valid month, which would actually be 1390, since 09/31 is a valid date.

but of course, a question like this wouldn't be complete without an *aha* factor. and of course, there are not 31 days in september, only 30. so we have to go back to august 08 which means the correct date would be **08/31/1380**.

palindromes also offer another great string question.
write a function that tests for palindromes
`bool isPalindrome( char* pStr )`

if you start a pointer at the beginning and the end of the string and keep comparing characters while moving the pointers closer together, you can test if the string is the same forwards and backwards. notice that the pointers only have to travel to the middle, not all the way to the other end (to reduce redundancy).

```
bool isPalindrome( char* pStr )
{
  if ( pStr == NULL )
   return false;

  char* pEnd = pStr;
  while ( *pEnd != '\0' )
    pEnd++;
  pEnd--;

  while(pEnd > pStr)
  {
    if ( *pEnd != *pStr )
      return false;

    pEnd--;
    pStr++;
  }

  return true;
}
```

---

problem: you are given a sequence of numbers from 1 to n-1 with one of the numbers repeating only once. (example: 1 2 3 3 4 5). how can you find the repeating number? what if i give you the constraint that you can't use a dynamic amount of memory (i.e. the amount of memory you use can't be related to n)?
what if there are two repeating numbers (and the same memory constraint?)

solution:

as a programmer, my first answer to this problem would be make a bit vector of size n, and every time you see the number, set its correspond index bit to 1. if the bit is already set, then that's the repeater. since there were no constraints in the question, this is an ok answer. its good because it makes sense if you draw it for someone, whether they are a programmer, mathemetician, or just your grandpa. its not the most efficient answer though.

now, if i add the constraint that you can only use a fixed amount of memory (i.e. not determined by n) and it must run in O(n) time... how do we solve it. adding all the numbers up from 1 to n-1 would give us a distinct sum. subtracting the total sum of all the numbers from the sum of n to n-1 (which is (n)(n-1)/2) would give us the secret extra number.

what if you can only use a fixed amount of memory, and **two** of the numbers are repeated? we know that the numbers have a distinct sum, and the difference would be equal to the sum of our unknowns

```
c = a + b
```

where c is the sum and *a* and *b* are the unknowns - c is a constant
if we had another similar formula we could solve the two unknown equations. my first thought was that the numbers would have a distinct product - (n-1)!
if we divide the total product by the (n-1)! product, we would get another equation

```
c2 = ab
```

we could then solve the two equations to get them into quadratic formula notation

```
0 = ax^2 + bx + c
```

and solve for the two values of x. this answer is correct but factorial grows really fast.

some sort of sum would be better. the sum of the squares from n-1 to 1 would work. that would yield a function of the form

```
c2 = a^2 + b^2
```

which could also be solved by using the quadratic equation.

i think its fine to remind someone of the quadratic equation... (maybe only because i myself had to look it up to solve the problem) i mean really though, the last time i used it was probably in 10th grade. as long as they get the idea that given two unknowns and two equations you can solve for the unknowns - thats the point.

$$\boxed{x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}}$$

---

five pirates have 100 gold coins. they have to divide up the loot. in order of seniority (suppose pirate 5 is most senior, pirate 1 is least senior), the most senior pirate proposes a distribution of the loot. they vote and if at least 50% accept the proposal, the loot is divided as proposed. otherwise the most senior pirate is executed, and they start over again with the next senior pirate. what solution does the most senior pirate propose? assume they are very intelligent and extremely greedy (and that they would prefer not to die).

five pirates have 100 gold coins. they have to divide up the loot. in order of seniority (suppose pirate 5 is most senior, pirate 1 is least senior), the most senior pirate proposes a distribution of the loot. they vote and if at least 50% accept the proposal, the loot is divided as proposed. otherwise the most senior pirate is executed, and they start over again with the next senior pirate. what solution does the most senior pirate propose? assume they are very intelligent and extremely greedy (and that they would prefer not to die).

(to be clear on what 50% means, 3 pirates must vote for the proposal when there are 5 for it to pass. 2 if there are 4. 2 if there are 3. etc...)

solution: most of the time i get people who give answers like "the most senior pirate takes half and divides the rest up among the least senior pirates." um, you missed the whole point to begin with. sorry.

any answer without a specific logic behind it is invalid. if i ask you why pirate 5 gave x coins to pirate 1, please don't say "because he's nice".

now for the real solution. pirate 5 being the most senior knows that he needs to get 2 other people to vote for his solution in order for him not to be executed. so who can he get to vote for him, and why would they choose to vote for him? if you start thinking that pirate 4 will never vote for him, because he would rather have 5 die and then be in charge and take it all for himself, you are on the right track. but it gets more complicated.

lets consider if there were only 1 pirate. obviously he would take it all for himself and no one would complain.

if there were 2 pirates, pirate 2 being the most senior, he would just vote for himself and that would be 50% of the vote, so he's obviously going to keep all the money for himself.

if there were 3 pirates, pirate 3 has to convince at least one other person to join in his plan. so who can he convince and how? here is the leap that needs to be made to solve this problem. **pirate 3 realizes that if his plan is not adopted he will be executed and they will be left with 2 pirates**. he already knows what happens when there are 2 pirates as we just figured out. pirate 2 takes all the money himself and gives nothing to pirate 1. so pirate 3 proposes that he will take 99 gold coins and give 1 coin to pirate 1. pirate 1 says, well, 1 is better than none, and since i know if i don't vote for pirate 3, i get nothing, i should vote for this plan.

now we know what happens when there are 3 pirates. so what happens with 4? well pirate 4 has to convince 1 other person to join in his plan. he knows if he walks the plank then pirate 3 will get 99 coins and pirate 1 will get 1 coin. pirate 4 could propose giving pirate 1 two coins, and surely pirate 1 would vote for him, since 2 is better than 1. but as greedy as he is, pirate 4 would rather not part with 2 whole coins. he realizes that if he gets executed, then pirate 3's scenario happens and pirate 2 gets the shaft in that scenario (he gets zero coins). so pirate 4 proposes that he will give 1 coin to pirate 2, and pirate 2 seeing that 1 is better than 0 will obviously vote for this plan.

a common objection is that pirate 2 is not guaranteed to vote for this plan since he might hope for the case when there are only 2 pirates and then he gets all the booty. but that is why i said that the pirates are extremely intelligent. pirate 2 realizes that pirate 3 is smart enough to make the optimal proposal, so he realizes that there will never be 2 pirates left, because 3 doesn't want to die and we just showed that 3 has a winning proposal.

so lets sum up at this point

```
Pirate 1  2  3  4  5
    5. ?  ?  ?  ?  ?
    4. 0  1  0 99  -
    3. 1  0 99  -  -
    2. 0 100 -  -  -
    1.100
```

once you see the pattern it becomes very clear. you have to realize that when a pirate's plan does not succeed then that means you are in the same situation with one less pirate.
1. pirate 1 needs 0 other people to vote for him. so he votes for himself and takes all the money. 2. pirate 2 needs 0 other people to vote for him. so he votes for himself and takes all the money. pirate 1 gets 0. 3. pirate 3 needs 1 other person to vote for him. he gives 1 coin to pirate 1 for his vote - if we are reduced to 2 pirates, pirate 1 gets 0 so pirate 1 knows 1 is better than none. pirate 3 takes 99. pirate 2 gets 0. 4. pirate 4 needs 1 other person to vote for him. he gives 1 coin to pirate 2 - if we reduce to 3 pirates, pirate 2 gets 0 so pirate 2 knows 1 is better than none. pirate 4 takes 99. pirate 3 gets 0. pirate 1 gets 0. 5. pirate 5 needs 2 other people to vote for him. its clear now that the 2 people he needs to convince are the 2 who get shafted in the 4 pirate scenario - pirate 3 and pirate 1. so he can give them each 1 coin (which is better than 0 - what they would get otherwise) and keep 98 for himself.

```
Pirate 1  2  3  4  5
    5. 1  0  1  0 98
```

what happens if there are 15 pirates? pirate 15 needs 7 other people to vote for him, so he recruits pirates 13,11,9,7,5,3, and 1 with 1 coin each and keeps 93 coins himself. those pirates will all vote for him because they know that they get 0 coins if he dies and pirate 14 is in charge.

hope you enjoyed this one. its my favorite interview question of all. it really allows the candidate to ask a lot of interesting questions and its really amazing when they reach the solution all by themselves (as all fogcreek employees have done so far).

---

100 fogcreek programmers are lined up in a row by an assassin. the assassin puts red and blue hats on them. they can't see their own hats, but they can see the hats of the people in front of them. the assassin starts in the back and says "what color is your hat?" the fogcreek programmer can only answer "red" or "blue." the programmer is killed if he gives the wrong answer; then the assassin moves on to the next programmer. the programmers in front get to hear the answers of the programmers behind them, but not whether they live or die. they can consult and agree on a strategy before being lined up, but after being lined up and having the hats put on, they can't communicate in any way other than those already specified. what strategy should they choose to maximize the number of programmers who are guaranteed to be saved?

problem: 100 fogcreek programmers are lined up in a row by an assassin. the assassin puts red and blue hats on them. they can't see their own hats, but they can see the hats of the people in front of them. the assassin starts in the back and says "what color is your hat?" the fogcreek programmer can only answer "red" or "blue." the programmer is killed if he gives the wrong answer; then the assassin moves on to the next programmer. the programmers in front get to hear the answers of the programmers behind them, but not whether they live or die. they can consult and agree on a strategy before being lined up, but after being lined up and having the hats put on, they can't communicate in any way other than those already specified. what strategy should they choose to maximize the number of programmers who are guaranteed to be saved?

this is a very difficult problem to solve during an interview (especially if you've already taxed the candidate's brain). look for obvious solutions first, and the reasoning behind them and then try to lead them to the ultimate solution.

a logical answer could be all the programmers would just say "red" and that way about half of them would survive on average, assuming the hats were distributed randomly.

this is a good start and should naturally lead to having every other programmer say the color of the hat in front of them. the first programmer would say the color of the hat in front of him, then the next programmer would just say that color that was just said. so we can guarantee that half survive - the even numbered programmers (since the person behind them told them the answer). and potentially if the hats were distributed randomly some of the programmers would get lucky and the hat in front of them would be the same color as their own. so this strategy should save more than half, and on average 75% of them would live.

at this point, if the solution is not clear, the candidate may give answers like, "they could agree that if they said their hat color in a soft voice, it means the hat in front of them is the same color, and if they say it in a loud voice, it means the hat in front is a different color". this is definitely good and on the correct track. another option is they could say "reeeeeeeeeeed" for x number of seconds, where x represented the distribution of hats where a hat was a bit in a binary number, (red = 1, blue = 0). another interesting answer. there are many others like these that "bend" the rules and come to a solution.

but the real solution acknowledges that the programmers can only say "red" or "blue" and cannot alter their voice in such a convincing way as to signal any information other than the word they said. a good way to get this point across, is simply to change the problem slightly by saying "the assassin gets to hear their plan before she puts the hats on, and so will try to thwart the plan however she can."

so if they decide to all say "red", she'll put blue hats on all of them. if they decide to all say the color of the hat in front of them, she'll alternate the hats on every head, guaranteeing half will die. even with the assassin hearing their plan, there is still a way to save almost everyone.

we know that the first person is never going to have any information about the color of their hat, so they cannot be guaranteed to survive. but, i'll give you a hint to the solution: i can save every other person for sure.

solution: they agree that if the number of red hats that the back person can see is even, that programmer will say "red". if they add up to an odd number, they will say "blue". this way number

99 can look ahead and count the red hats. if they add up to an even number and number 100 said "red", then 99 must be wearing a blue hat. if they add up to an even number and number 100 said "blue", signalling an odd number of red hats, number 99 must also be wearing a red hat. number 98 knows that 99 said the correct hat, and so uses that information along with the 97 hats in front to figure out what color hat is on 98's head.

sample:

```
100  99  98  97  96  95  94 ... facing ->
 R    B   B   R   B   R   B ... -> 45 R and 48 B
```

this shows #100 wearing a red hat, 99 a blue, 98 a blue, 97 a red, 96 a blue, 95 a red, 94 a blue and 45 red hats - 48 blue hats on the people in front of them.

100 counts up the red hats: 47 total. so 100 says "blue". the assassin kills 100. 99 counts up the red hats in front: 47. 100 said blue, so 100 saw an odd number. 99 sees an odd number, so 99 says "blue" and lives. 98 had counted 47 red hats, and 99 didn't say "red" so thats still the total. 98 says "blue". 97 counts up and finds 46 red hats. 99 and 98 didn't say "red", so his count is missing a red hat (its on his head, he realizes). he says "red". 96 heard the "red" and now knows that there are an even number of "red" hats in front of 95. 96 sees 46, so he knows he has a "blue" hat. etc...

even if the assassin knows the plan, she can't thwart it. she hears the plan, but she still has to put the hats on their heads. the plan doesn't rely on any ordering of the hats, so the worst the assassin can do is to make sure #100 gets killed and thats the worst damage she can do.

---

a bad king has a cellar of 1000 bottles of delightful and very expensive wine. a neighbouring queen plots to kill the bad king and sends a servant to poison the wine. (un)fortunately the bad king's guards catch the servant after he has only poisoned one bottle. alas, the guards don't know which bottle but know that the poison is so strong that even if diluted 1,000,000 times it would still kill the king. furthermore, it takes one month to have an effect. the bad king decides he will get some of the prisoners in his vast dungeons to drink the wine. being a clever bad king he knows he needs to murder no more than 10 prisoners - believing he can fob off such a low death rate - and will still be able to drink the rest of the wine at his anniversary party in 5 weeks time.

explain how....

problem: a bad king has a cellar of 1000 bottles of delightful and very expensive wine. a neighbouring queen plots to kill the bad king and sends a servant to poison the wine. (un)fortunately the bad king's guards catch the servant after he has only poisoned one bottle. alas, the guards don't know which bottle but know that the poison is so strong that even if diluted 1,000,000 times it would still kill the king. furthermore, it takes one month to have an effect. the bad king decides he will get some of the prisoners in his vast dungeons to drink the wine. being a clever bad king he knows he needs to murder no more than 10 prisoners - believing he can fob off such a low death rate - and will still be able to drink the rest of the wine at his anniversary party in 5 weeks time.

explain how...

solution: i'll give you a hint. 1000 is less than 1024. if there were 1024 or more bottles of wine it would take more than 10 prisoners.

number the bottles 1 to 1000, and write the number in binary format.

```
bottle 1    = 0000000001
bottle 250  = 0011111010
bottle 1000 = 1111101000
```

now take your prisoner's 1 through 10 and let prisoner 1 take a sip from every bottle that has a 1 in its least significant bit. let prisoner 10 take a sip from every bottle with a 1 in its most significant bit. etc.

```
prisoner    10 9 8 7 6 5 4 3 2 1
bottle 924   1 1 1 0 0 1 1 1 0 0
```

for instance, bottle #924 would be sipped by 10,9,8,5,4 and 3. that way if bottle #924 was the poisoned one, only those prisoners would die.

after four weeks, line the prisoners up in their bit order and read each living prisoner as a 0 bit and each dead prisoner as a 1 bit. the number that you get is the bottle of wine that was poisoned.

additional question: to increase your chance of living, which prisoner would you want to be?

if there were 1023 bottles, it wouldn't matter since everyone would have to take 512 sips. but there are 23 bottles less, so the people whose bits would have been on from 1001 to 1023 won't have to take a sip. 1001 is [11111 01001] in binary and 1023 is [11111 11111]. the most five significant bits are the most interesting because they would always be on from 1001 to 1023, so all those people are missing out on 23 bottles of wine that they otherwise would have had to drink. so in order to increase your chance of living, you'd probably want to be prisoner 6 to 10. (but depending on how the king determines who is least significant and who is most significant you could get shafted.)

note that if the king was really trying to kill the least number of prisoners, he should have let 999 prisoners each take a sip from their respective bottle numerically (if he had that many prisoners at his disposal). that way only one prisoner would die, and there's a chance of 1/1000 that no one would die, but then the puzzle isn't very fun.

---

you have three jars that are all mislabeled. one contains peanut butter jelly beans, another grape jelly jelly beans, and the third has a mix of both (not necessarily a 50/50 mix, could be a 1/99 mix or a 399/22 mix). how many jelly beans would you have to pull out, and out of which jars, to find out how to fix the labels on the jars?

```
|     |       |     |         |     |
|jar 1|       |jar 2|         |jar 3|
|     |       |     |         |     |
=======       =======         =======
  p.b.         grape        p.b./grape
```

you have three jars that are all mislabeled. one contains peanut butter jelly beans, another grape jelly jelly beans, and the third has a mix of both (not necessarily a 50/50 mix, could be a 1/99 mix or a 399/22 mix). how many jelly beans would you have to pull out, and out of which jars, to find out how to fix the labels on the jars?

```
|     |       |     |         |     |
|jar 1|       |jar 2|         |jar 3|
|     |       |     |         |     |
=======       =======         =======
  p.b.         grape        p.b./grape
```

solution: 1 jelly bean from the p.b./grape jar will do the trick.

the trick here is to realize that every jar is mislabeled. therefore you know that the peanut butter jelly bean jar is not the penut butter jelly bean jar, and the same goes for the rest.

you also need to realize that it is the jar labeled p.b./grape, labelled as the mix jar, that is your best hope. if you choose a jelly bean out of there, then you will know whether that jar is peanut butter or grape jelly jelly beans. it can't be the mix jar because i already said that every jar is mislabeled.

once you know that jar 3 is either peanut butter, or grape jelly, then you know the other jars also. if it is peanut butter, then jar 2 must be mixed because it can't be grape (as its labelled) and it can't be peanut butter (that's jar 3). hence jar 1 is grape.

if jar 3 is grape, then you know jar 1 must be the mix because it can't be p.b. (as its labelled) and it can't be grape (that's jar 3). hence jar 2 is peanut butter.

if you pick jelly beans from jar 1 or jar 2, then you would have to pick out **all** of the jelly beans before you knew what that jar was. this is because jar 1 and 2 could be the mix, so in order to disprove that they were the mix, you would have to pull out **every** jelly bean just to make sure (since there could just be one bean of the opposite flavor in there).

---

problem: this one is a classic that many of you have probably already heard, but all the more reason why it should definitely be included here. four people are on this side of the bridge. the bridge will be destroyed by a bomb in 17 minutes. everyone has to get across before that. problem is that it's dark and so you can't cross the bridge without a flashlight, and they only have one flashlight. plus the bridge is only big enough for two people to cross at once. the four people walk at different speeds: one fella is so fast it only takes him 1 minute to cross the bridge, another 2 minutes, a third 5 minutes, the last it takes 10 minutes to cross the bridge. when two people cross the bridge together (sharing the flashlight), they both walk at the slower person's pace. can they all get across before the bridge blows up?

```
person A: 1 minute
person B: 2 minutes
person C: 5 minutes
person D:10 minutes
```

problem: four people are on this side of the bridge. the bridge will be destroyed by a bomb in 17 minutes. everyone has to get across before that. problem is that it's dark and so you can't cross the bridge without a flashlight, and they only have one flashlight. plus the bridge is only big enough for two people to cross at once. the four people walk at different speeds: one fella is so fast it only takes him 1 minute to cross the bridge, another 2 minutes, a third 5 minutes, the last it takes 10 minutes to cross the bridge. when two people cross the bridge together (sharing the flashlight), they both walk at the slower person's pace. can they all get across before the bridge blows up?

```
person A: 1 minute
person B: 2 minutes
person C: 5 minutes
person D:10 minutes
```

solution: of course its possible, otherwise it wouldn't be a very interesting question. the only trick is in realizing that you want to get the two slowest people across together, because otherwise you are wasting too much time. but then once you get them across, how do you not make one of them walk back with the flashlight? well, you just have one of the fast people already there waiting to sprint the flashlight back across.

```
1. A & B cross.  total time:  2 minutes.

 C |==========================| A
 D |                          | B
   |==========================| flashlight

2. B comes back. total time:  4 minutes.

 C |==========================| A
 D |                          |
 B |==========================|
flashlight
```

```
3. C & D cross.  total time: 14 minutes.

 B |==========================| A
   |                          | C
   |==========================| D
                               flashlight

4. A comes back. total time: 15 minutes.

 A |==========================| C
 B |                          | D
   |==========================|
flashlight

5. A & B cross.  total time: 17 minutes.

   |=========    =============| A
   |       KABOOM!            | B
   |=========    =============| C D
                              flashlight
```

another solution which is valid is to have A bring the flashlight back in step 2. it only changes the solution slightly. this is supposed to be a "classic" microsoft interview question but it seems absurdly easy to be a good interview question (especially coupled with the fact that everyone has probably heard it already).

---

this is a card trick without the trick. there is no sleight of hand, no tricks up my sleeve, no magic whatsoever. it's all done with logic, yet it will amaze most people.

joel and i are working together as a team to do the trick. babak will be the culprit.

i ask babak to pick 5 cards out of a deck. he can pick any five cards, he can shuffle the deck 7 times, it really doesn't matter. he honestly picks out 5 cards that i cannot see. he hands the five cards to me (joel can't see any of this). i look at the cards and i pick 1 card out and give it back to babak. i then arrange the other four cards in a special way, and give those 4 cards all face down, and in a neat pile, to joel. joel looks at the 4 cards i gave him, and says out loud which card babak is holding (suit and number).

i did not convey any information to joel other than the way i ordered the 4 cards, (all face down, aligned in line with one another) so how did i encode babak's card using that method?

hint 1: card trick without the trick
hint 2: card trick without the trick

hint 1: with five cards there will always be at least two cards of the same suit. since i get to pick the card to give back to babak, i can use that to my advantage to communicate the suit to joel.

hint 2: if i communicate the suit to joel with one of the cards, then i'm only left with the other 3 cards to show the number. since only the ordering of the cards can be used, i am allowed up to six combinations. well, that won't do to signify up to 13 cards, will it? certainly not, but if i pick the right card to show the suit, i can use that card to reduce the range of cards down to 6. how?

solution: card trick without a trick

this is a card trick without the trick. there is no sleight of hand, no tricks up my sleeve, no magic whatsoever. it's all done with logic, yet it will amaze most people.

joel and i are working together as a team to do the trick. babak will be the culprit.

i ask babak to pick 5 cards out of a deck. he can pick any five cards, he can shuffle the deck 7 times, it really doesn't matter. he honestly picks out 5 cards that i cannot see. he hands the five cards to me (joel can't see any of this). i look at the cards and i pick 1 card out and give it back to babak. i then arrange the other four cards in a special way, and give those 4 cards all face down, and in a neat pile, to joel. joel looks at the 4 cards i gave him, and says out loud which card babak is holding (suit and number).

i did not convey any information to joel other than the way i ordered the 4 cards, (all face down, aligned in line with one another) so how did i encode babak's card using that method?

hint 1: card trick without the trick
hint 2: card trick without the trick

solution: hint 1 talks about how there are guaranteed to be at least 2 cards with the same suit in a series of five cards. i use this to signal to joel the suit of the card. i have to make a decision though between which of the two cards to give back to babak. which one do i pick?

hint 2 talks about how if i use one of the cards to signal the suit, that only leaves me with 3 cards. i can only arrange 3 cards in six permutations. how do i signal a number between 2 and 13 using only six permutations?

simple really. consider that the series of cards is a cycle.

```
2 3 4 5 6 7 8 9 10 J Q K A 2 3 4 5 6 7 8 9 10 J Q K A 2 ... etc
```

i take the two cards of the same suit (if there are more than two of the same suit, then i choose any two, it won't affect the solution) and i pass babak the one that is further to the right in the cycle. if i do it correctly, the one further to the right will be no more than 6 places away than the one to the left. (it just depends on how you think about the cycle).

for example. say we had the 2 and the 9 of hearts. if you look at it as the 2 to the left and the 9 to the right, it is seven cards away. but if you look as the 9 to the left and the 2 to the right in the cycle, the 2 is only 6 cards away.

then i just come up with a system of the permutations that signals how many places to the right to count to find the correct card

one such system could be:
sort the 3 non-same suit cards first by number then by suit (diamonds, spades, hearts, clubs) to give a unique sort for every card.
label the cards as A, B, and C.
place the cards in the following order to signal how many places away babak's card is:

```
A B C - 1
A C B - 2
B A C - 3
B C A - 4
C A B - 5
C B A - 6
```

then put the suit card on top and give the pile to joel.

example:
babak picks out the following five cards:

```
3C 7D 10S QH AD
```

the 7 of diamonds (7D) and the ace of diamonds (AD) are the two suit cards. i recognize that AD is going to be the left card and 7D the right (because the other way they are 7 spaces apart). i give babak back the 7D. i sort the other cards, 3C 10S QH. i have to signal 6 places to the right (7D - AD = 6). so i place them in C B A format: QH 10S 3C. i then place the AD on top and pass to joel.

joel pulls off the first card and sees that it is the AD. (he knows babak's card is a diamond). he looks at the next 3 cards and realizes their ordering is C B A. signalling 6, he counts 6 spots from the ace. (2 3 4 5 6 7). then he says "7 of diamonds" and the crowd is amazed.

another **sneakier card trick** (with the trick) works as follows.

babak picks one card out of the deck and shows it to me. he doesn't tell joel and i don't speak. after seeing the card, i sneakily use my one hand to signal both the number and suit of the card. i use my four fingers (not the thumb) as bits signalling a binary number. if they are extended it means 1, if i don't extend them it means 0. the pinky is the least significant. so to signal 7 i would extend all fingers except my index finger. the suit is signalled by four positions of the thumb (hidden, adjacent to the hand, slightly ajar, fully extended). i can use this method to tell joel the card without saying anything (and making any noticeable motions). with some practice it can really amaze people (at least it worked with my family).

---

a slightly different version of the original pirates problem (read that one first to get all the rules). 6 pirates, only one gold coin. as before, the pirates are super-smart, and they value, in this order: (i) their lives, (ii) getting money, (iii) seeing other pirates die. so if given the choice between two outcomes, in which they get the same amount of money, they'd choose the outcome where they get to see more of the other pirates die. how can pirate 6 save his skin?

problem: a slightly different version of the original pirates problem (read that one first to get all the rules). 6 pirates, only one gold coin. as before, the pirates are super-smart, and they value, in this order: (i) their lives, (ii) getting money, (iii) seeing other pirates die. so if given the choice between two outcomes, in which they get the same amount of money, they'd choose the outcome where they get to see more of the other pirates die. how can pirate 6 save his skin?1 pirate case is obvious. 2 pirate case is obvious, pirate 2 keeps the coin.

3 pirate case: pirate 3 has to give the coin to pirate 1, because if he gives it pirate 2 pirate 2 will say "screw that i wanna see you die and i'm going to get the coin anyway." so in 3 pirate case, we have pirate 1 gets 1 coin, pirates 2 and 3 get 0.

4 pirate case: pirate 4 can't give the coin to pirate 1, because pirate 1 would rather see him die since he's going to get 1 coin anyway. But pirate 4 could give the coin to either pirate 2 or pirate 3.

5 pirate case: pirate 5 just dies, and it goes to the 4 pirate case. there is no way for him to convince two people to vote for him.

6 pirate case: pirate 6 can count on his own vote, plus pirate 5's vote, because 5 won't want to die. now who should he give the coin to? he could give it to pirate 1 or to pirate 4, since in the 4 pirate case they are guaranteed to get nothing. it's unclear whether he could give it to pirate 2 or 3. neither pirate 2 nor pirate 3 is guaranteed to get a coin in the 4 pirate case. so the question is, how do they value (i) definitely getting a coin from pirate 6 vs. (ii) definitely seeing pirates 6 and 5 die, with the chance of getting a coin from pirate 4. since we don't have enough information to answer that, to be safe, i would just say pirate 6 should offer the coin to pirate 1 or pirate 4.

---

a mad bomber is out on the job, making bombs. he has two fuses (pieces of string) of varying thickness which each burn for 30 seconds. unfortunately he wants this bomb to go off in 45 seconds.

he can't cut the one fuse in half because the fuses are different thicknesses and he can't be sure how long it will burn. how can he arrange the fuses to make his bomb go off at the right time?

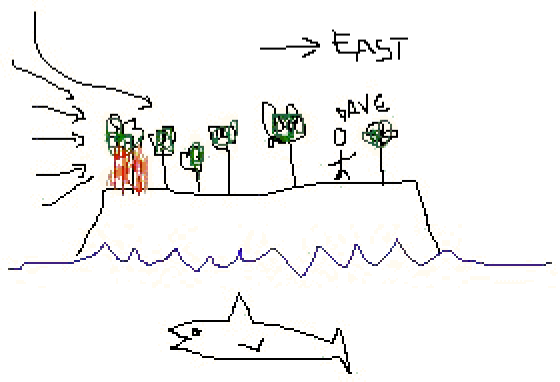
solution: fuse on fire

problem: a mad bomber is out on the job, making bombs. he has two fuses (pieces of string) of varying thickness which each burn for 30 seconds. unfortunately he wants this bomb to go off in 45 seconds. he can't cut the one fuse in half because the fuses are different thicknesses and he can't be sure how long it will burn. (for example: the first half of the fuse might burn up in 10 seconds and the second half in 20 seconds.. the fuse doesn't burn at a constant rate, but the total time it would burn is 30 seconds). how can he arrange the fuses to make his bomb go off at the right time?

solution: light both ends of one of the fuses. when that fuse goes out, 15 seconds has elapsed. then light the other fuse.

---

dave winer is stuck on a deserted island, with lots of trees, which is very thin and ten miles long (east to west). large cliffs surround the entire island and if he jumped off, he wouldn't survive the fall. a fire starts burning at the west side of the island. unfortunately this island always has a west to east blowing wind blowing at 2mph and this moves the fire slowly toward dave at 1mph. (so he only has ten hours left). save dave (or maybe, let him burn :-) ! what to do?



solution: dave's on fire

someone suggested he could dig a pit across the island to act as a firebreak. good suggestion, if he had a shovel and the ground wasn't too hard.

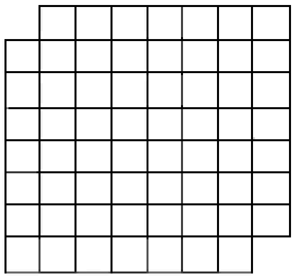
but even if he didn't have a shovel, he could pick up a branch and run up to the fire and light the branch. then run all the way to the eastern edge of the island, but stop about a mile short. there he could light all those trees on fire and they would start burning and the fire would move east. it would consume all that vegetation in an hour, and then dave could wait for awhile for that part to cool down some. when the initial fire reached him, he could just run into the already burnt part, and the fire couldn't get him.

```
T = tree
D = dave
B = burnt

top view
==========================
| fire-> T T T T T T T D |
==========================

dave gets branch and lights it from fire
==========================
| fire-> D T T T T T T T |
==========================

dave lights trees farther east on the island
==========================
| fire-> T T T T D fire->|
==========================

dave waits until second fire cools, and then hides out there
==========================
| B B B fire-> T T B B D |
==========================
```

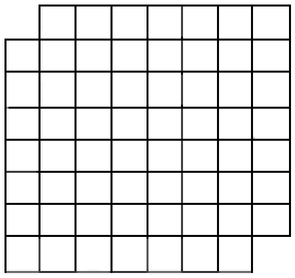
dave is saved!

---

problem: using 31 dominoes, where one domino covers exactly two squares, can you cover all the empty squares on this chessboard (which has 62 spaces). if so, how? if not, why?



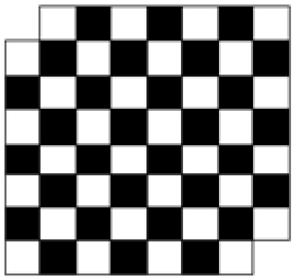
problem: using 31 dominoes, where one domino covers exactly two squares, can you cover all the empty squares on this chessboard (which has 62 spaces). if so, how? if not, why?



solution: i think everyone's first inclination is to try and figure out how it is possible. then again, if you've heard a bunch of these questions before, you usually know that if the question says "if not, why?" or "prove whether its possible or impossible", you can infer that it is not possible (otherwise, the question usually just asks for the solution).

after awhile fiddling around with the dominoes, you will start to realize that the most obvious solutions are not viable. if you start to think its impossible, think about why.

a reader emailed me telling me that using the picture of the chessboard without the squares colored will result in a very small percentage of people solving the problem. if i use the following picture



a much higher percentage of people will be able to solve the problem. the people who solve the problem using only the grid, usually represent the problem in their heads as the colored board.

still can't figure it out? using the colored board, remember that a domino always covers a black square and a white square.

if you look at the board, you will see that the two squares missing are both black. this means that for all the squares on the board, each white square has a corresponding black square, except for two white ones. so even if you covered all the black/white pairs, you would still be left with two white squares, which will never be adjacent to each other, no matter how you lay out the dominoes.

three cannibals and three anthropologists have to cross a river. the boat they have is only big enough for two people. if at any point in time there are more cannibals on one side of the river than anthropologists, the cannibals will eat them. what plan can the anthropologists use for crossing the river so they don't get eaten?

remember! the boat can't cross the river by itself, someone has to be in it to row it across.

a much harder river crossing problem will appear later this week.

```
A - anthropologist
C - cannibal
++ - boat

        river
AAA |============|
    |++          |
CCC |============|

need to make it
        river
    |============| AAA
    |          ++|
    |============| CCC
```

note that if you violate the "anthropologists > cannibals" rule at any point in time, it is illegal.. for example if a boat with a cannibal and an anthropologist travels to a shore with one cannibal on it, then # cannibals > # anthropologists, even if you say the anthropologist immediately takes the boat back.

problem: three cannibals and three anthropologists have to cross a river. the boat they have is only big enough for two people. if at any point in time there are more cannibals on one side of the river than anthropologists, the cannibals will eat them. what plan can the anthropologists use for crossing the river so they don't get eaten?

```
A - anthropologist
C - cannibal
++ - boat

        river
AAA |============|
    |++          |
CCC |============|

need to make it
        river
    |============| AAA
    |          ++|
    |============| CCC
```

note that if you violate the "anthropologists > cannibals" rule at any point in time, it is illegal.. for example if a boat with a cannibal and an anthropologist travels to a shore with one cannibal on it, then # cannibals > # anthropologists, even if you say the anthropologist immediately takes the boat back.

Let W be the west shore which they are all on. Let E be the east shore where they want to go.

```
1. A and C cross
W = { A, A, C, C }
E = { A, C }

2. A returns
W = { A, A, A, C, C }
E = { C }

3. C and C cross
W = { A, A, A }
E = { C, C, C }

4. C returns
W = { A, A, A, C }
E = { C, C }

5. A and A cross
W = { A, C }
E = { A, A, C, C }

6. A and C return
W = { A, A, C, C }
E = { A, C }

7. A and A cross
W = { C, C }
E = { A, A, A, C }

8. C returns
W = { C, C, C }
E = { A, A, A }
```

```
9. C and C cross
W = { C }
E = { A, A, A, C, C }

10. C returns
W = { C, C }
E = { A, A, A, C }

11. C and C cross
W = true
E = { A, A, A, C, C, C }
```

---

part I: draw a square. divide it into four identical squares. remove the bottom left hand square. now divide the resulting shape into four identical shapes.

part II: draw an equilateral triangle (all sides same length). divide it into four identical shapes. remove the bottom left hand shape. now divide the resulting shape into four identical shapes.

this is the sort of problem that i would expect on a MENSA test. i'm not too sure whether getting this right constitutes intelligence in a way that would benefit computer scientists, but maybe it does. if you figure it out, then you can say it does. if you can't figure it out, then you can just say it's all hogwash and it's a stupid question.

thanks to mark chesser

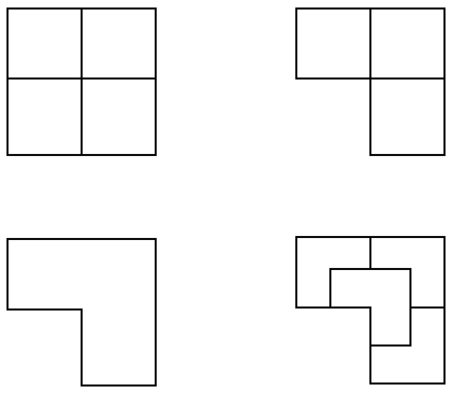
part I: draw a square. divide it into four identical squares. remove the bottom left hand square. now divide the resulting shape into four identical shapes.

part II: draw an equilateral triangle (all sides same length). divide it into four identical shapes. remove the bottom left hand shape. now divide the resulting shape into four identical shapes.

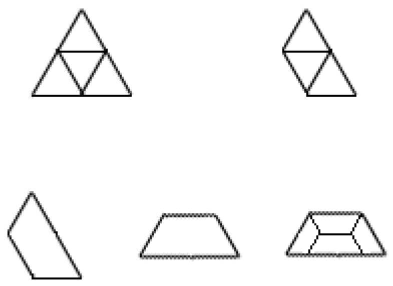
this is the sort of problem that i would expect on a MENSA test. i'm not too sure whether getting this right constitutes intelligence in a way that would benefit computer scientists, but maybe it does. if you figure it out, then you can say it does. if you can't figure it out, then you can just say it's all hogwash and it's a stupid question.

solution:

part I



part II



---

five webloggers - joshua Allen, meg Hourihan, jason Kottke, robert Scoble, and joel Spolsky - were competing for karma points on the major search engines: google, yahoo, altavista, lycos, and msn. karma was distributed on a five point scale. the most popular weblog received 5 points, and the least popular received 1 point. for each search engine, no two webloggers received the same number of points. overall scores were determined by adding up the individual scores from each search engine.

Allen got the highest number of karma points - 24. Kottke was consistent in his scores: he got the same karma points from 4 different search engines. Spolsky got 5 points from lycos, and 3 from msn.

no two webloggers got the same total score, and the final rankings were as follows: Allen, Hourihan, Kottke, Scoble, and Spolsky. how many karma points did Hourihan get from lycos?

five webloggers - joshua Allen, meg Hourihan, jason Kottke, robert Scoble, and joel Spolsky - were competing for karma points on the major search engines: google, yahoo, altavista, <>lycos, and msn. karma was distributed on a five point scale. the most popular weblog received 5 points, and

the least popular received 1 point. for each search engine, no two webloggers received the same number of points. overall scores were determined by adding up the individual scores from each search engine.

Allen got the highest number of karma points - 24. Kottke was consistent in his scores: he got the same karma points from 4 different search engines. Spolsky got 5 points from lycos, and 3 from msn.

no two webloggers got the same total score, and the final rankings were as follows: Allen, Hourihan, Kottke, Scoble, and Spolsky. how many karma points did Hourihan get from lycos?

solution: let's start with what we know

```
    G   Y   A   L   M    Total
   |============================
 A |                       24
 H |
 K |
 Sc|
 Sp|            5   3
```

the only possible values for Allen achieving 24 is { 5 5 5 5 4 } and since Spolsky got a 5 from lycos, we know that is where Allen's 4 comes from.

we also know that the total number of points given out was 75.
```
(5 * (5 + 4 + 3 + 2 + 1))
```

spolsky had to have at least 11 points. if Spolsky had more than 11 points, say 12, then is it possible to achieve a solution? Scoble would have had to have at least 13 (since there were no ties), and Kottke 14, and Houlihan 15. that would yield an overall total of 78. too much! so Spolsky definitely had 11 points.

```
    G   Y   A   L   M    Total
   |============================
 A | 5   5   5   4   5      24
 H |
 K |
 Sc|
 Sp| 1   1   1   5   3      11
```

using the same logic as before, we also know that Scoble could not have gotten more than 12 points. if he had 13, and Kottke 14, and Houlihan 15, the total would be 77. still too much. so Scoble had 12, and continuing on Kottke had to have 13 and Houlihan 15, otherwise the totals would be over 75.

now we know Kottke had 14 points. if he got four 4's for consistency, it wouldn't work (already over 16). if he got four 2's, it also wouldn't work (8 points plus the maximum 5 is still only 13). so he had to have received four 3's. and since he couldn't have gotten a 3 from msn, that is where he received a 1.

	G	Y	A	L	M	Total
A	5	5	5	4	5	24
H						15
K	3	3	3	3	1	13
Sc						12
Sp	1	1	1	5	3	11

let's look at scoble. we can see from the chart that all 5's and 3's have already been given out (and there is only one 1 left). so Scoble's scores can only contain 4's, 2's, or a single 1. given that information the only possible combination of 5 scores that would yield 12 is { 2 2 2 2 4 }. since Allen already has a 4 from lycos, Scoble must have a 2 there.

	G	Y	A	L	M	Total
A	5	5	5	4	5	24
H						15
K	3	3	3	3	1	13
Sc				2		12
Sp	1	1	1	5	3	11

hence Houlihan must have a 1 from lycos!

	G	Y	A	L	M	Total
A	5	5	5	4	5	24
H				**1**		15
K	3	3	3	3	1	13
Sc				2		12
Sp	1	1	1	5	3	11

---

a disfunctional family has to cross the river. on one side of the river are a mom and 2 daughters, dad and 2 sons, the maid and the dog. there is a boat only big enough to hold 2 people (counting the dog as 1 person). only the adults are capable of operating the boat. everyone has to get to the other side, without anything bad happening.

difficulties: if the dog is left with anyone and the maid isn't there to control him, he'll bite. the dad can't be left with any of the daughters when the mom isn't there. likewise, the mom can't be trusted alone with either of the sons when the dad isn't there.

remember! only an adult can operate the boat, AND the boat can't drive itself.

disfunctional family has to cross the river. on one side of the river are a mom and 2 daughters, dad and 2 sons, the maid and the dog. there is a boat only big enough to hold 2 people (counting the dog as 1 person). only the adults are capable of operating the boat. everyone has to get to the other side, without anything bad happening.

difficulties: if the dog is left with anyone and the maid isn't there to control him, he'll bite. the dad can't be left with any of the daughters when the mom isn't there. likewise, the mom can't be trusted alone with either of the sons when the dad isn't there.

remember! only an adult can operate the boat, AND the boat can't drive itself.

solution:
we start with a mother (m), two daughters (d1, d2), a father (f), two sons (s1, s2), a housemaid (h), and a dog (c - canine) on the west (W) shore, and they all want to get to the east (E) shore.

```
W = {m, d1, d2, f, s1, s2, h, c} // everyone on the west shore
E = {} // no one on the east shore
```

let's move everyone, over...

housemaid and canine go east, and the housemaid comes back:

```
W = {m, d1, d2, f, s1, s2, h}
E = {c}
```

housemaid and s1 go east, h and c come back:

```
W = {m, d1, d2, f, s2, h, c}
E = {s1}
```

father and s2 go east, father comes back:

```
W = {m, d1, d2, f, h, c}
E = {s1, s2}
```

mother and father go east, mother comes back:

```
W = {m, d1, d2, h, c}
E = {f, s1, s2}
```

h and c go east, father comes back:

```
W = {m, d1, d2, f}
E = {s1, s2, h, c}
```

father and mother go east, mother comes back:

```
W = {m, d1, d2}
E = {f, s1, s2, h, c}
```

mother and d1 go east, housemaid and c come back:

```
W = {d2, h, c}
E = {m, d1, f, s1, s2}
```

h and d2 go east, h comes back

```
W = {h, c}
E = {m, d1, d2, f, s1, s2}
```

h and c go east

```
W = {}
E = {m, d1, d2, f, s1, s2, h, c}
```

done!

---

this is a classic problem which i have heard many times before. this is the "harder" of the two problems, since in this one, you do not know if the invalid item weighs more or less than the others.

solving it is only half the battle. writing up a solution that anyone including your grandma could understand, is very hard.

problem: the evil king from before sends his own assassin to take care of the evil queen who tried to poison him. of course, her trusty guards catch the assassin before any harm is done. the queen notices that the assassin is quite handsome and doesn't really want to punish him by death. she decides to test his wisdom.

the queen gives the assassin 12 pills which are all completely identical in shape, smell, texture, size, except 1 pill has a different weight. the queen gives the man a balance and tells him that all the pills are deadly poison except for the pill of a different weight. the assassin can make three weighings and then must swallow the pill of his choice. if he lives, he will be sent back to the bad king's kingdom. if he dies, well, thats what you get for being an assassin.

only one pill is not poison and it is the pill which has a different weight. the assassin does not know if it weighs more or less than the other pills. how can he save his skin?

this is a classic problem which i have heard many times before. this is the "harder" of the two problems, since in this one, you do not know if the invalid item weighs more or less than the others.

solving it is only half the battle. writing up a solution that anyone including your grandma could understand, is very hard.

problem: the evil king from before sends his own assassin to take care of the evil queen who tried to poison him. of course, her trusty guards catch the assassin before any harm is done. the queen notices that the assassin is quite handsome and doesn't really want to punish him by death. she decides to test his wisdom.

the queen gives the assassin 12 pills which are all completely identical in shape, smell, texture, size, except 1 pill has a different weight. the queen gives the man a balance and tells him that all the pills are deadly poison except for the pill of a different weight. the assassin can make three weighings and then must swallow the pill of his choice. if he lives, he will be sent back to the bad king's kingdom. if he dies, well, thats what you get for being an assassin.

only one pill is not poison and it is the pill which has a different weight. the assassin does not know if it weighs more or less than the other pills. how can he save his skin?

solution: easy.

choose any eight of the pills and put four of them on each side of the balance.

there are two possibilities:

(1) one side of the balance comes out lighter. In this case, you know that the abnormal (safe) pill is one of the pills already on the balance. label the pills on the lighter side A B C and D, and the pills on the heavier side E F G and H. label the pills not on the balance NORM (you know they’re normal pills).

(2) the balance is even. in this case, you know that the abnormal (safe) pill is one of the pills not on the balance. label the pills already on the balance NORM, and label the four pills not on the balance I J K and L.

let’s proceed with possibility (1).

consider why the side ABCD came out higher than the side EFGH. this could be because:

A is the abnormal pill, and it’s lighter than the other pills.

B is the abnormal pill, and it’s lighter than the other pills.

C is the abnormal pill, and it’s lighter than the other pills.

D is the abnormal pill, and it’s lighter than the other pills.

E is the abnormal pill, and it’s heavier than the other pills.

F is the abnormal pill, and it’s heavier than the other pills.

G is the abnormal pill, and it’s heavier than the other pills.

H is the abnormal pill, and it’s heavier than the other pills.

now let’s make another weighing, with two of the ABCD pills on either side, and one of the EFGH pills on either side. for example, let’s weigh ABE versus CDF. how would this weighing come out given each of those 8 possibilities we just listed?

if A is the light pill, the ABE/CDF weighing will come out with ABE high.

if B is the light pill, the ABE/CDF weighing will come out with ABE high.

if C is the light pill, the ABE/CDF weighing will come out with ABE low.

if D is the light pill, the ABE/CDF weighing will come out with ABE low.

if E is the heavy pill, the ABE/CDF weighing will come out with ABE low.

if F is the heavy pill, the ABE/CDF weighing will come out with ABE high.

if G is the heavy pill, the ABE/CDF weighing will come out even.

if H is the heavy pill, the ABE/CDF weighing will come out even.

OK, so we observe how the ABE versus CDF weighing actually comes out.

(a) if it comes out even, then we know that the abnormal pill is either G or H. for our third weighing, we can weigh G against one of the pills we already know to be normal (one of the pills we labelled NORM). if it comes out even, then G is normal and H must be the abnormal pill. if it comes out uneven, then G is the abnormal pill.

(b) as we can see from our chart above, if the ABE/CDF weighing comes out with ABE high, then the situation is either: A is the light pill, B is the light pill, or F is the heavy pill.

(c) as we can see from our chart above, if the ABE/CDF weighing comes out with ABE low, then the situation is either: C is the light pill, D is the light pill, or E is heavy pill.

so in either situation (b) or (c), we have two possible light pills and one possible heavy pill. what we do in that case is we put one of the possible light pills and the possible heavy pill on one side of the scale, and two NORM pills on the other side of the scale. this is our third weighing. if it comes out even, then we know that the other possible light pill is the abnormal pill. if it comes out with the two NORM pills high, then we know that one of the pills on the other side is abnormally heavy, so we know that the possible heavy pill is the culprit. if it comes out with the two NORM pills low, then we know that one of the pills on the other side is abnormally light, so we know that the possible light pill on the scale is the culprit.

that takes care of case (1), where the first weighing came out uneven.

what about case (2), where the first weighing comes out even?

then we know the abnormal pill is one of I J K or L, and we have two weighings to find the abnormal pill in.

for our second weighing, we put I and J on one side of the scale, and two NORM pills on the other.

(a) if this comes out uneven, we know the abnormal pill is I or J; we weigh I against one NORM pill to see if I is abnormal and if it isn’t, we can conclude that J is the abnormal pill.

(b) if the IJ versus 2 NORM weighing comes out even, we know the abnormal pill is K or L; we weight K against one NORM pill to see if K is abnormal and if it isn’t, we can conclude that L is the abnormal pill.

finished.

---

another well known problem in probability is the monty hall problem.

you are presented with three doors (door 1, door 2, door 3). one door has a million dollars behind it. the other two have goats behind them. you do not know ahead of time what is behind any of the doors.

monty asks you to choose a door. you pick one of the doors and announce it. monty then counters by showing you one of the doors with a goat behind it and asks you if you would like to keep the door you chose, or switch to the other unknown door.

should you switch? if so, why? what is the probability if you don't switch? what is the probability if you do.

lots of people have heard this problem.. so just knowing what to do isn't sufficient. its the explanation that counts!

another well known problem in probability is the monty hall problem.

you are presented with three doors (door 1, door 2, door 3). one door has a million dollars behind it. the other two have goats behind them. you do not know ahead of time what is behind any of the doors.

monty asks you to choose a door. you pick one of the doors and announce it. monty then counters by showing you one of the doors with a goat behind it and asks you if you would like to keep the door you chose, or switch to the other unknown door.

should you switch? if so, why? what is the probability if you don't switch? what is the probability if you do.

lots of people have heard this problem.. so just knowing what to do isn't sufficient. its the explanation that counts!

the answer is that yes, you should *always* switch as switching increases your chances from 1/3 to 2/3. how so, you ask? well, lets just enumerate the possibilities.

	door 1	door 2	door 3
case 1	$$	goat	goat
case 2	goat	$$	goat
case 3	goat	goat	$$

its clear that if you just choose a door and stick with that door your chances are 1/3.

using the switching strategy, let's say you pick door 1. if its case 1, then you lose. if it's case 2, monty shows you door 3, and you switch to door 2, you win. if it's case 3, monty shows you door 2, and you switch to door 3, you win. it doesn't matter what door you pick in the beginning, there are always still three possibilities. one will cause you to lose, and two will cause you to win. so your chances of winning are 2/3.

the solution all resides in the fact that monty knows what is behind all the doors and therefore always eliminates a door for you, thereby increasing your odds.

maybe its easier to see in this problem. there are 1000 doors, only one of which has a prize behind it. you pick a door, then monty opens 998 doors with goats behind them. do you switch? it seems more obvious in this case, because monty had to take care in which door not to open, and in the process basically showing you where the prize was (999 out of 1000 times).

---

a man has a gold chain with 7 links. he needs the service of a laborer for 7 days at a fee of one gold link per day. however, each day of work needs to be paid for separately. in other words, the worker must be paid each day after working and if the laborer is ever overpaid he will quit with the extra money. also he will never allow himself to be owed a link.

what is the fewest # of cuts to the chain to facilitate this arrangement and how does that guarantee payment?

Can we get change back from the laborer?

If so, we cut one link to make a chain of 4 links, a chain of 2 links and the cut link itself.

Day 1, we give him the cut link
Day 2, we take back the cut link, give him the 2 link chain
Day 3, we give him the cut link
Day 4, we take back both the cut link and the 2 link chain, give him the 4 link chain
Day 5, we give him the cut link
Day 6, we take back the cut link, give him the 2 link chain
Day 7, we give him the cut link

---

a one armed surgeon with a hand wound needs to operate on three patients. the surgeon only has two gloves. how can he operate on the three patients in turn without risking exchange of fluids? (remember he only has one arm so he only needs to wear one glove at a time.)

problem: a one armed surgeon with a hand wound needs to operate on three patients. the surgeon only has two gloves. how can he operate on the three patients in turn without risking exchange of fluids? (remember he only has one arm so he only needs to wear one glove at a time.)

solution: the surgeon places both gloves on his hand (1 and 2). he operates on patient A. he then takes the top glove off (#2), leaving on the bottom glove (#1) and operates on patient B. then he carefully reverses glove #2, so the clean side is on the outside, and he places it on top of glove #1 which is on his hand, and operates on patient C.

this problem is kind of dumb because how's the surgeon going to change the gloves on his hand when he only has one hand. plus no offense, but how often do you come across a one-armed surgeon (i'm sure there are plenty of one-armed doctors, but a surgeon!?!). anyway, i had to make this problem child friendly and changing the story to the above was the only way to do it. consider for a minute what the initial problem was. the surgeon was just a guy, the patients were women, and the glove was... well, i won't insult your intelligence.

---

part I: what is the angle between the minute hand and the hour hand at 3:15 on an analog clock? no, its not 0.

part II: how often does the minute hand pass the hour hand on an analog clock?

part I: what is the angle between the minute hand and the hour hand at 3:15 on an analog clock? no, its not 0.

part II: how often does the minute hand pass the hour hand on an analog clock?

answer: part I: 12 hours on the clock make 360 deg. so one hour is 30 deg. the hour hand will be directly on the 3 when the minute hand is at 12 (3:00). after 15 minutes or 1/4 of an hour, the hour hand will be 1/4 * 30 deg = 7.5 deg. away from the minute hand.

part II: if you just think about it, intuitively you'll see the minute hand passes the hour hand 11 times every 12 hours, so it must pass it every 1 1/11 hours. but this doesn't make sense to me. i need to prove it.

if x is our answer then every x hours, the minute hand and the hour hand will be right on top of each other. every hour the hour hand travels 5 units. so between every time that the minute and the hour hand meet, the hour hand will go 5*x units. every hour the minute hand travels 60 units, so it will have gone 60*x units.

what we're trying to find is the distance traveled by the minute hand to reach the hour hand, once the minute hand has looped around once. consider its 12:00. both hands in the same position. after an hour, minute hand is on 12, hour hand on 1 (its traveled 5 units). now in the time it takes the minute hand to catch up to the hour hand it will travel a little bit further.

we only need to find x where 5*x = 60*(x-1), since the real distance traveled by the minute hand, from where it started to where it ends, is 60*(x-1). the first hour just puts it back where it started, so we're only concerned with the extra part it traveled to reach the hour hand.

```
5x = 60(x-1)
5x = 60x - 60
60 = 55x
60/55 = x
```

there it is. the answer is 60/55 hours, or every 1 and 1/11 hours.

i apologize that this is horribly confusing, but if you stare at it long enough it will make sense.

---

at 6 a.m. a man starts hiking a path up a mountain. he walks at a variable pace, resting occasionally, but never actually reversing his direction. at 6 p.m. he reaches the top. he camps out overnight. the next morning he wakes up at 6 a.m. and starts his descent down the mountain. again he walks down the path at a variable pace, resting occassionally, but always going downhill. at 6 p.m. he reaches the bottom. what is the probability that at some time during the second day, he is in the exact same spot he was in on the first day?

problem: at 6 a.m. a man starts hiking a path up a mountain. he walks at a variable pace, resting occasionally, but never actually reversing his direction. at 6 p.m. he reaches the top. he camps out overnight. the next morning he wakes up at 6 a.m. and starts his descent down the mountain. again he walks down the path at a variable pace, resting occassionally, but always going downhill. at 6 p.m. he reaches the bottom. what is the probability that at some time during the second day, he is in the exact same spot he was in on the first day?

answer: the probability is 100%. the easiest way to see it is, consider that on the second day when the man is going down the mountain, a ghost follows his original pace up the mountain. so even if he varies his pace as he goes down the mountain, at some point in time, he will be in the same spot as the ghost, and therefore, the same spot he was in the day before.

---

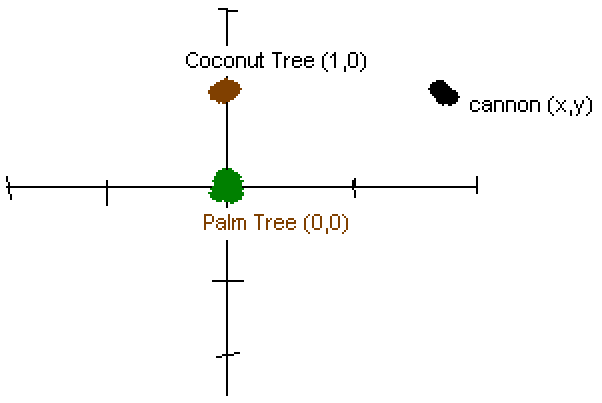
you find an old treasure map in your grandma's attic. the map shows a cannon, a coconut tree, and a palm tree. the map states that to find the treasure you must:
a. start at the cannon, walk toward the palm tree while counting your paces. when you reach the palm tree, turn 90 degrees to your **left** and walk the same number of paces. mark that spot on the ground with a stake.
b. start at the cannon again, walk toward the coconut tree while counting your steps. when you reach the coconut tree, turn 90 degrees to your **right** and walk the same number of paces. mark that spot on the ground with a stake.
c. find the midpoint between the two stakes and dig for the treasure.

you set off in secrecy to the deserted island. upon reaching the shore you site the coconut tree and the palm tree, but someone has removed the cannon. without digging randomly all over the island, is it still possible to find the treasure?

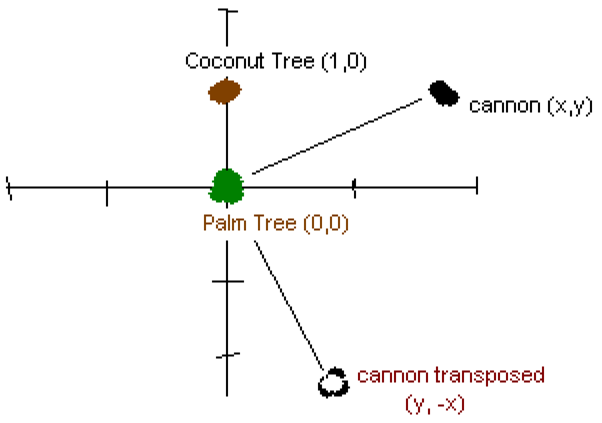
problem: you find an old treasure map in your grandma's attic. the map shows a cannon, a coconut tree, and a palm tree. the map states that to find the treasure you must:
a. start at the cannon, walk toward the palm tree while counting your paces. when you reach the palm tree, turn 90 degrees to your **left** and walk the same number of paces. mark that spot on the ground with a stake.
b. start at the cannon again, walk toward the coconut tree while counting your steps. when you reach the coconut tree, turn 90 degrees to your **right** and walk the same number of paces. mark that spot on the ground with a stake.
c. find the midpoint between the two stakes and dig for the treasure.

you set off in secrecy to the deserted island. upon reaching the shore you site the coconut tree and the palm tree, but someone has removed the cannon. without digging randomly all over the island, is it still possible to find the treasure?

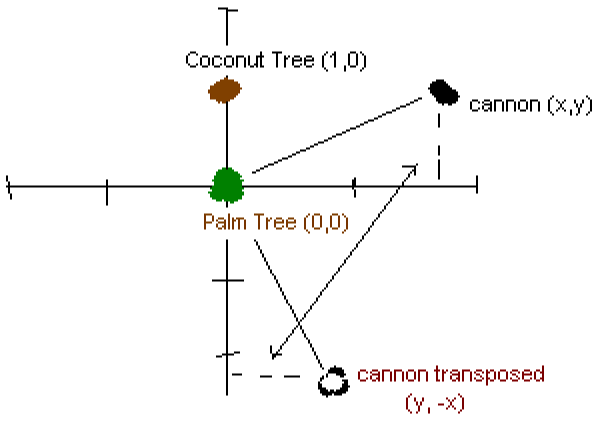
solution: this just takes basic geometry skills. when we get to the island all we see are the coconut and palm trees. so lets lay out our coordinate system such that the palm tree is at (0,0) and the coconut tree is at (1,0). it honestly doesn't matter how you describe the coordinate system - you could say the coconut is at (c,0) if you like, or even (0,c) or (0,1). we are just placing our own coordinate system on top of the existing surface. if you use a different system, you will get a different numerical answer but the same positional answer in the real world.



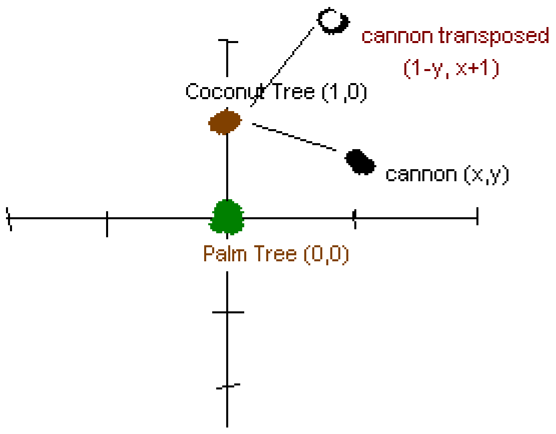
here is our island. the cannon is at (x,y) because we have no idea where it is, so x and y are the unknowns. (note the cannon doesn't have to be in the upper right quadrant but it won't make a difference in the solution because x and y could be negative if we want them to be).



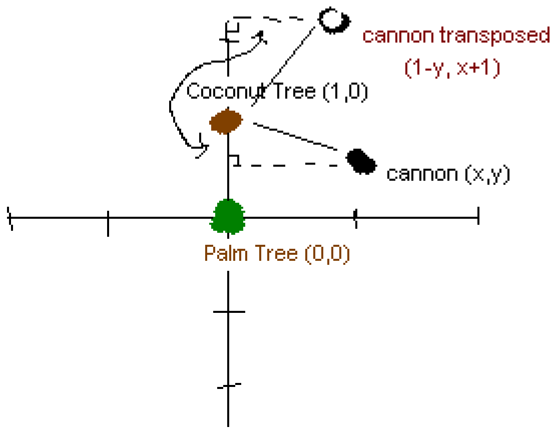
if we walk to the palm and turn right we can easily see the way it lays out in the diagram. you basically have just transposed the x and y positions.



it might be easier to think of it as a triangle, and flipping the triangle around to find the point for the next stake at (y, -x)



we do the same thing with the coconut tree, although here its only a tiny bit trickier because we have to factor in the position of the coconut tree at (1,0) or (c,0).



we use the idea of a triangle to help us better understand that the stake will end up at (1-y, x+1) or (c-y, x+c).

then to find the midpoint of the two points which is just the first position + the second position divided by two.

```
(y, -x) and (1-y, x+1)
((y - y + 1)/2, (x - x + 1)/2)
(1/2, 1/2)
```

hence our answer is (1/2,1/2) - although we'll see that if we had use the constant C for the coconut tree we would have ended up with (c/2, c/2). this is important because even though we laid out the island in our own coordinate system, its not always the case that c must be positive. i think (and i've seen this solution elsewhere, but can't really come up with a repro, so let me know if i'm wrong here) there really are two places the treasure could be because c could also be negative. so if we use our first answer of (1/2,1/2) we must also search at (-1/2, -1/2). if the cannon was in the lower half of the coordinate system you can see that the treasure would actually be in that quadrant also

---

a. first i would just like to say i am so glad jenna got voted off survivor. yuck, i couldn't stand her ever since she accused that guy of eating beef jerky when he was chewing on grass.
b. recruit meyer on boot camp rocks. did you see how he made himself cry to get sympathy? and it worked! he is the funniest part of the show. i hope he wins.
c. i apologize for this site being **soooooo slow**. but that's why you get what you pay for. (i didn't pay anything so i shouldn't really expect anything in return.)
d. those screwy pirates are at it again. this time there are 13 pirates and they need to protect their treasure chest. they decide that they should only be able to open the chest if the majority (at least 7) agree that it should be opened. they ask a locksmith to come and put a specific number of locks on the safe. every lock must be opened to open the chest. there can be multiple keys for each lock, but each key only opens one lock (i.e. no skeleton keys). the locksmith can give more than one key

to each pirate. how many locks should the locksmith use and what strategy should he use to distribute the keys, such that only when a majority of the pirates agree can the chest be opened?

problem: those screwy pirates are at it again. this time there are 13 pirates and they need to protect their treasure chest. they decide that they should only be able to open the chest if the majority (at least 7) agree that it should be opened. they ask a locksmith to come and put a specific number of locks on the safe. every lock must be opened to open the chest. there can be multiple keys for each lock, but each key only opens one lock (i.e. no skeleton keys). the locksmith can give more than one key to each pirate. how many locks should the locksmith use and what strategy should he use to distribute the keys, such that only when a majority of the pirates agree can the chest be opened?

a somewhat technical answer is here.

(un)fortunately that user left it up to me to explain why this solution works... but fortunately my brother did it here in a very understandable fashion. thanks big brother!

Here is solution that works for any number of pirates and any number of pirates needed to open to chest.

Let T be the total number of pirates. Let N be the number of pirates required to open the chest.

The number of locks needed would be L = (T,N-1) = (T!)/[(N-1)! * (T-N-1)!]. The number of keys each pirate would have be K = (T-1,N-1) = (T-1)!/[(N-1)! * (T-N)!].

For this specific problem, the number of locks would be 1716 and the number of keys per pirate would be 924.

I know I haven't provided an explanation of why and how this system works. We'll leave that lengthy, involved explanation to the author.

Note: the notation (x,y) is the combination notation; I can't use the conventional combination notation in plain text. (x,y) basically asks the question "how many ways can you pick y objects from a group of x objects?" (x,y) = (x!)/[y! * (x-y)!]

Here's a less elegant solution. How many different combinations of 6 pirates are there? 13 choose 6 is: 13!/6!7! That's a hell of a lot of locks. But you could say, put that many locks on the chest. And for each lock, you distribute keys to all the pirates EXCEPT the selection of 6 corresponding to that lock. Then any selection of 7 pirates is guaranteed to be open all the locks. For consider the first 6 of them. There will be exactly ONE lock that those six are unable, between the 6 of them, to open. But the seventh pirate will be able to open that lock. Because keys to that lock went out to everybody except those 6.

Also, any selection of 6 pirates is guaranteed to encounter exactly one lock that they are unable to open. (So any selection of fewer than 6 is guaranteed to encounter 1 or more locks they are unable to open...)

So that will do the trick, and the explanation of why it would do the trick is elegant. But I don't feel that it's an elegant solution, because of there being so many locks... Anyone have something better

---

there are three ants on a triangle, one at each corner. at a given moment in time, they all set off for a different corner at random. what is the probability that they don't collide?

Consider the triangle ABC. We assume that the ants move towards different corners along the edges of the triangle.

Total no. of movements: 8 A->B, B->C, C->A A->B, B->A, C->A A->B, B->A, C->B A->B, B->C, C->B A->C, B->C, C->A A->C, B->A, C->A A->C, B->A, C->B A->C, B->C, C->B

Non-colliding movements: 2 A->B, B->C, C->A A->C, B->A, C->B

(i.e. the all ants move either in the clockwise or anti-clockwise direction at the same time)

P(not colliding) = 2/8 = 0.25

---

how many places are there on the earth that one could walk one mile south, then one mile east, then one mile north and end up in the same spot? to be precise, let's assume the earth is a solid smooth sphere, so oceans and mountains and other such things do not exist. you can start at any point on the sphere and walk in any direction you like.

think you've figured it out? i'll tell you now, there is more than one. in fact, there are more than two. also be advised that walking north from the north pole (or south from the south pole) is illogical and therefore does not enter into the problem. all normal assumptions about directions will be used.

there are **no** tricks involved with this question. it just forces you to really think about the problem to come up with all the solutions.

problem: how many places are there on the earth that one could walk one mile south, then one mile east, then one mile north and end up in the same spot? to be precise, let's assume the earth is a solid smooth sphere, so oceans and mountains and other such things do not exist. you can start at any point on the sphere and walk in any direction you like. think you've figured it out? i'll tell you now, there is more than one. in fact, there are more than two. also be advised that walking north from the north pole (or south from the south pole) is illogical and therefore does not enter into the problem. all normal assumptions about directions will be used.

there are no tricks involved with this question. it just forces you to really think about the problem to come up with all the solutions.

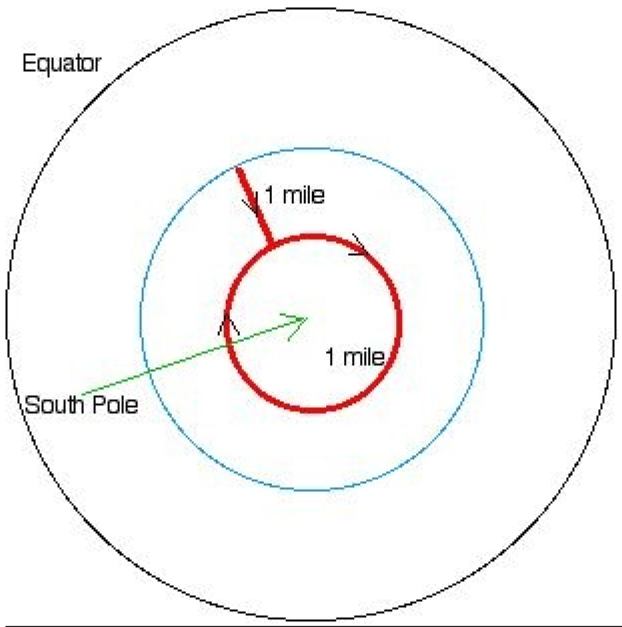
solution:

well the north pole is one such place.

then somewhere near the south pole such that when you walk one mile south you are at the point on the earth where the circumference is 1. that way when you walk 1 mile east, you end up back at the same point. and of course one mile north from there puts you back where you started. here is a drawing courtesy of jy. there may or may not be such a place in the northern hemisphere where walking a mile south puts you at the 1 mile circumference point on the earth.

i'm no geometry sphere expert, so someone will have to let me know if that is physically possible (i.e. i tend to think that if you walk n units south from any point on the northern part of a sphere, other than the north pole, it is impossible for the circumference to be n or less than n, but who knows?)

finally there are actually an infinite number of points. if we consider the case before where we went to the point with a circumference of 1, why not go to the point with a circumference of 1/2. then when you go a mile east, you loop around twice, and end up in the same spot. this holds true for 1/3, 1/4, 1/5, ... 1/n, etc.



---

this is difficult to describe in words, so read this carefully, lest there be any confusion. you have a normal six sided cube. i give you six different colors that you can paint each side of the cube with (one color to each side). how many different cubes can you make?

different means that the cubes can not be rotated so that they look the same. this is important! if you give me two cubes and i can rotate them so that they appear identical in color, they are the same cube.

Let X be the number of "different" cubes (using the same definition as in the problem). Let Y be the number of ways you can "align" a given cube in space such that one face is pointed north, one is east, one is south, one is west, one is up, and one is down. (We're on the equator.) Then the total number of possibilities is X * Y. Each of these possibilities "looks" different, because if you could take a cube painted one way, and align it a certain way to make it look the same as a differently painted cube aligned a certain way, then those would not really be different cubes. Also note that if you start with an aligned cube and paint it however you want, you will always arrive at one of those X * Y possibilities.

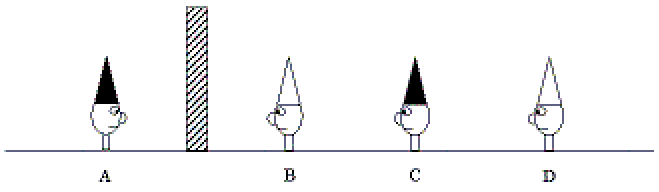
How many ways can you paint a cube that is already "aligned" (as defined above)? You have six options for the north side, five options for the east side, etc. So the total number is 6! (that's six factorial, or 6 * 5 * 4 * 3 * 2 * 1). Note that each way you do it makes the cube "look" different (in the same way the word is used above). So 6! = X * Y.

How many ways can you align a given cube? Choose one face, and point it north; you have six options here. Now choose one to point east. There are only four sides that can point east, because the side opposite the one you chose to point north is already pointing south. There are no further options for alignment, so the total number of ways you can align the cube is 6 * 4.

Remember, Y is defined as the number of ways you can align the cube, so Y = 6 * 4. This gives us 6! = X * 6 * 4, so X = 5 * 3 * 2 * 1 = 30.

---

buried four fishermen up to their necks in the sand on the beach at low tide for keeping their fishing spot a secret from me. i put a hat on each of their heads and told them that one of them must shout out the correct color of their own hat or they will all be drowned by the incoming tide. i give them 10 minutes to do this. fisherman A and B can only see the sand dune i erected. fisherman C can see that fisherman B has a white hat on. fisherman D can see that C has a black hat, and B has a white hat. the fisherman have been told that there are four hats, two white and two black, so they know that they must have either a white or a black hat on. who shouts out the color of their hat and how do they know?



Fisherman C shouts out.

Fishermen A and B are in the same situation - they have no information to help them determine their hat colour so they can't answer. C and D realise this.

Fisherman D can see both B and C's hats. If B and C had the same colour hat then this would let D know that he must have the other colour.

When the time is nearly up, or maybe before, C realises that D isn't going to answer because he can't. C realises that his hat must be different to B's otherwise D would have answered. C therefore concludes that he has a black hat because he can see B's white one.

---

you die and the devil says he'll let you go to heaven if you beat him in a game. the devil sits you down at a round table. he gives himself and you a huge pile of quarters. he says "ok, we'll take turns putting quarters down, no overlapping allowed, and the quarters must rest on the table surface. the first guy who can't put a quarter down loses." the devil says he wants to go first.

being the smart programmer you are, you realize that if the devil goes first, he may automatically win. so you convince him to let you go first, which makes your day because you know you can't lose. what is your winning strategy?

Here's what I came up with:

First, put the first quarter exactly in the center of the (perfectly circular) table.

Next, for each quarter the opponent places, place one directly opposite it. That is, place it so that the center of the table is halfway between your piece and the opponent's previous piece.

This will generate a completely symettric (about the center) layout of quarters on the table. This means that whenever the opponent selects a free space to place a quarter in, the space opposite is guaranteed to be free as well. Since we are always guaranteed an open space, we will never lose with this strategy (and thus win when there are finally no more spaces for the opponent to use).

---

there is a pot of N noodles. (so there are 2N ends). a person randomly grabs two ends and merges them. the person keeps doing it, until there are no more noodles, (and only loops), left in the pot. what's the average number of loops in the pot?

OK, all the answers so far just list formulae, without giving any explanation. I'll try to work it out out loud. (At this point I have no idea what the answer is.) Also, it's a math problem, so I'll assume we can ignore factors like how much the noodles stick together, how stiff they are, and so on. I'm sure I have no idea to take account of those factors.

Call the first end he picks up Noodle i. The second end he picks up is Noodle i*.

When he sticks the first two ends together, there are two possible outcomes:

(a) i* is the other end of noodle i, so he has created a loop.

(ii) i* is a different noodle from i, so he has created one long noodle out of two.

What are the odds of (a) happening? There are (2n-1) ends left in the bowl once he picks up the end of noodle i, and only 1 of them is the other end of the same noodle. Abstracting away from all physical details, let's say the odds of getting the other end of the same noodle are 1/(2n-1).

So the odds of (b) happening are 1-[1/(2n-1)], which is [(2n-1)-1]/(2n-1), i.e. (2n-2)/(2n-1).

If (a) happened, now we have a bowl with one loop in it, and n-1 other unlooped noodles. We add 1 to our count of loops, and repeat the problem for n-1.

If (b) happened, now we have a bowl with 0 loops in it, and n-1 other unlooped noodles. It's just that one of those noodles is especially long. We don't add anything to our count of loops; we just repeat the problem for n-1.

Now, when we get down to 1 unlooped noodle left in the bowl, the odds of (a) happening are 1.

The average # of loops will be: for each point where a loop could be formed, add 1 * the probability of a loop being formed then.

So we can write a function:

```
real average_number_of_loops (int n) {
  if (n == 1) {
   return 1
 } else {
   -- there is a 1/(2n-1) chance of getting 1 loop formed here
   -- and a (2n-2)/(2n-1) chance of getting 0 loops formed here
    -- and in either case we then have the same problem repeated for
    -- n-1 noodles
    -- so we should return ([(1/(2n-1))*1] + [(2/(2n-1))*0]) + average_number_of_loops(n-1)
    -- or, simplifying...
    return (1/(2n-1)) + average_number_of_loops(n-1)
  }
```

Equivalently:

```
real average_number_of_loops (int n) {
  if (n == 0) {
   return 0
  } else {
```

```
  return (1/(2n-1)) + average_number_of_loops(n-1)
 }
```

So I guess I agree with the guy who wrote:

> Summation for i = 1 to N of 1 / 2i - 1. Sorry I can't figure out how to > resolve.

Except that you have to understand his "1 / 2i - 1" as being "1/ (2i - 1)." Which is no doubt what he intended. But now we have an explanation of why.

---

a person dies, and arrives at the gate to heaven. there are three doors. one of them leads to heaven. another one leads to a 1-day stay at hell, and then back to the gate, and the other leads to a 2-day stay at hell, and then back to the gate. every time the person is back at the gate, the three doors are reshuffled. how long will it take the person to reach heaven?

this is a probability question - i.e. it is solvable and has nothing to do with religion, being sneaky, or how au dente the pasta might be ;-)

1/3 of the time, the door to heaven will be chosen, so 1/3 of the time it will take zero days. 1/3 of the time, the 1-day door is chosen; of those, the right door will be chosen the next day, so 1/9 trips take 1 day. Similarly, 1/9 will take two days (choosing the 2-day door, then the right door).

After that, the cases split again, and again, and again. I can't seem to make a nice infinite sum this way, so let's try again.

Suppose the average days spent is X. 1/3 of the cases are done in zero days as before. 1/3 of the cases are 1 day plus X. 1/3 are 2 + X. So:

```
X = 1/3 * 0 + 1/3 * (1 + X) + 1/3 * (2 + X)
  = 0 + 1/3 + X/3 + 2/3 + X/3
  = 1 + 2X/3
```

Therefore,

```
  X/3 = 1
    X = 3
```

On average, it takes three days to get to heaven. Two if the noodles are limp.

Took me one blind alley, and about five minutes. (heh heh)

---

someone walks into your room and dumps a huge bag of quarters all over the floor. they spread them out so no quarters are on top of any other quarters. a robot then comes into the room and is programmed such that if it sees a head, it flips it to tails. if it sees a tail, it throws it in the air. the robot moves around randomly forever. will there be a convergence in distribution of heads vs. tails?

Hmmm. I made a little math trick out of it. If the 'bot finds a head, it flips. If the bot finds a tail, there's a fifty percent chance this will become a head, as well.

(P_h = #heads/#coins, P_t = #tails/#coins)

So, delta h = -P_h + .5 P_t

= -(1 - P_t) + .5 P_t

= 1.5 P_t -1

Which is zero when P_t is 2/3. It's important to remember that a flip to a tail results in no change to the number of tails -- this threw me off for a second.

---

i challenge you to a game. we each get one penny and we flip them at the same time. (so on turn 1, we each flip our respective pennies - turn 2, we flip them again, and so on until someone wins). i am looking to get heads then tails. you are looking to get heads then heads. so if you flip heads on any flip and then heads on the next flip, you win. if i flip heads on any flip and then tails on the next flip, i win. (its not a speed race, we both flip at the same time, except i'm only concerned with what appears on my coin, and you are only concerned with whats on your coin). are the odds fair? (obviously not, otherwise this wouldn't be a question). who has the advantage and why?

In a 3 turn game:

O needs HH, X needs HT

HHH O

HHT OX

HTH X

HTT X

THH O

THT X

TTH

TTT

X wins 4 times out of 8, O wins 3 times out of 8

Its because in the event X loses there is a 50% chance of starting out on another H where O has to start on a T 75% of the time. (or something like that...)

---

how does one find a loop in a singly linked list in O(n) time using constant memory? you cannot modify the list in any way (and constant memory means the amount of memory required for the solution cannot be a function of n.)

I first figured this out when I was asked the question in a Microsoft interview, so there's verification that one question in the book was from a real interview. The best part of this problem is that I actually needed to write the code out for it a little while ago to detect a bug.

One way to detect a loop is to iterate over the list with 2 pointers at the same time where one is iterating at double speed. If the 2 pointers are ever equal after they iterate once and before they both reach an end, there's a loop.

Now for another puzzle.. I think the next question I had to answer was something along the lines of "OK, How do you remove a loop in a linked list with the same constraints?" The latter question definitely seems harder, but in the sequence of questions I'd already answered for the interviewer, the solution was pretty obvious. I'll leave that solution to someone else today.

Hm. I think I should pick up that book. I've always wondered where people got their juicy interview riddles.. I've always just repeated ones I've heard before or made up new ones along the same lines.

---

you have 20 blue balls and 14 red balls in a bag. you put your hand in and remove 2 at a time. if they're of the same color, you add a blue ball to the bag. if they're of different colors, you add a red ball to the bag. (assume you have a big supply of blue & red balls for this purpose. note: when you take the two balls out, you don't put them back in, so the number of balls in the bag keeps decreasing). what will be the color of the last ball left in the bag?

once you tackle that, what if there are 20 blue balls and 13 red balls to start with?

You always take off Red Balls two by two !

So if you start with 14 Red Balls, you cannot have one single Red ball at the end.

... so the last ball is blue.

But if you start with 13 Red Balls, as you take them off 2 by 2 (at one moment or the other you'll do it !) you will arrive at a moment when you have 1 red ball in the bag. But as you can only take off Red Balls 2 by 2 (Did I already say that ?!) you'll remove the last Blue Balls, one by one......

So the lastball will be red...

Oh, by the way, did I tell you that you take off Red Balls 2 by 2 ?! ;->

For tired people here is why you take off Red Balls 2 by 2 :
- If you take off 1 RED and 1 BLUE, in fact you will take off 1 BLUE
- If you take off 2 RED, in fact you will take off 2 RED (and add 1 BLUE)
- If you take off 2 BLUE, in fact you will take off 1 BLUE

---

every night, i dump all the change in my pocket into a big bucket.

when I buy things, i never hand over coins. always bills. so i accumulate a lot of coins. even if the purchase price is $1.01, and i have lots of coins in my pocket, i pay $2 and take the 99c in change. all the more coins to dump in my change bucket!

after about 10 years of this, i decide to roll all the coins into rolls. remember that a quarter roll is $10, a dime roll is $5, nickels $2, pennies 50 cents. so I go to the Banking Supply Store and buy empty paper rolls.

the Banking supply store, conveniently, sells assortment packs of coin rolls. each assortment pack contains W quarter rolls, X dime rolls, Y nickel rolls, and Z penny rolls.

the question: what is the optimum ratio of W to X to Y to Z to maximize the probability that I will use up the assortment packs at the same rate, e.g. without lots of leftover nickel tubes and stuff.

p.s. this problem should ideally be solved using Excel (but if you really like doing these things by hand, be my guest).

paul brinkley makes these assumptions which are all good assumptions to make:
Assumption 1: The price of purchases made, modulo $1, is an even distribution from 0 cents to 99 cents.
Assumption 2: The cashier will always give you the least number of coins mathematically possible, and will always have enough of each type of coin to do this. So you'll never get 99 pennies as change for a $1.01 purchase, for example.
Assumption 3: Half dollars don't exist.

"Brute force" approach:

I guess I'll begin with a few assumptions:

Assumption 1: The price of purchases made, modulo $1, is an even distribution from 0 cents to 99 cents. Probably not true, but we can cover that later.

Assumption 2: The cashier will always give you the least number of coins mathematically possible, and will always have enough of each type of coin to do this. So you'll never get 99 pennies as change for a $1.01 purchase, for example.

Assumption 3: Half dollars don't exist.

Over the long haul, then, you'd get N sets of 0 cents, 1 cent, 2 cents, and so on up to 99 cents. So let's consider one of each. How many of each coin would you end up with?

Easy first: let's do quarters. 25-49 cents each gets you one. 50-74 each gets you two. 75-99 each gets you three. That's (1+2+3)*25, or 150 quarters.

Dimes next. 10-19 gets you one. 20-24, two. (You'd get a quarter instead for 25 and up.) 35-44, one. 45-49, two. 60-69, 85-94, one. 70-74, 95-99, two. So that's 4*(10+5*2), or 80 dimes.

Nickels. One for 5-9, 15-19, and that same pattern for +25, +50, +75, so it's 4*10 or 40 nickels. You'll never get two at a time, since a dime (at least) will do.

Pennies. Let's cut to the chase: 1,2,3, and 4 cents gets 10 pennies in all, and that pattern happens 20 times, so that's 200 pennies.

Let's double-check. 200 pennies, 40 nickels, 80 dimes, 150 quarters. That's 200*1 + 40*5 + 80*10 + 150*25 cents = 200+200+800+3750 = 4950 cents, which is the sum of all numbers from 0 to 99, so it checks.

15/8/4/20 would be the ratio then, IF coin rolls all hold the same number of coins, but they don't. Quarter rolls hold 40 coins, dime rolls hold 50, nickel rolls 40, penny rolls 50. So you need 5/4 as many quarter and nickel rolls. The final ratio is 75/32/20/80. Seems like a lot of quarters and pennies, except for the assumption that you'll tend to get them much more often than nickels and dimes as change.

The numbers change slightly when you figure in things like frequency of coins in circulation, the supply the cashier has at any one time, the actual distribution of change values, and the cashier's inclination to give you two dimes and a nickel instead of a quarter just because...

So what numbers do assortment packs really contain?

---

a man has two cubes on his desk. every day he arranges both cubes so that the front faces show the current day of the month. what numbers are on the faces of the cubes to allow this?

Hmm. I don't know why this would warrant four aha's, but we'll see...

First, to show all possible days, we'd need one of each of the ten digits. We'd also need two 1s and two 2s to show 11 and 22. That's twelve numbers right there. Two cubes, twelve faces, so every face is used. Quite elegant.

We know each cube will need a 1 and a 2. Let's put the 3 on one of them. The 0 has to go on the other. We put 4, 5, and 6 on the 3-cube since we need to show 04 05 06.

But now where do 7, 8, and 9 go? The 0-cube needs them to be on the 3-cube, but it's full. I'm beginning to see why this gets four aha's.

Let's try this: clear both cubes, put the 0 on one of them. Now 1-9 have to go on the other cube to show 01-09. We can't put a 0 on the other cube, too, because that puts us over the 12-digit limit. It seems I have mathematical proof that this cannot be done! What am I missing? It's not like you can turn one of the other numbers sidewise to make another 0...

Heh.

You CAN make a 9 out of a 6, though. That frees up a digit. So you put 0, 1, and 2 on both cubes. Put 3 on one of them. 4 and 5 can go on it too. Put 6, 7, and 8 on the other. Now you can show 01-31 with no problem, and even 00 and 32 if you're feeling weird.

I'd've given it two, maybe three aha's. :)

---

in a country in which people only want boys, every family continues to have children until they have a boy. if they have a girl, they have another child. if they have a boy, they stop. what is the proportion of boys to girls in the country?

Pretty simple. Half the couples have boys first, and stop. The rest have a girl. Of those, half have a boy second, and so on.

So suppose there are N couples. There will be N boys. There will be an "infinite" sum of girls equal to N/2 + N/4 + N/8 + ... As any college math student knows, this sum adds up to N. Therefore, the proportion of boys to girls will be pretty close to 1:1, with perhaps a few more boys than girls because the sum isn't actually infinite.

Btw, hope you enjoyed your fishing trip, Michael!

---

i flip a penny and a dime and hide the result from you. "one of the coins came up heads", i announce. what is the chance that the other coin also came up heads?

Assuming complete honesty on the part of the flipper, wouldn't the solution be 33%?

There are four possible scenarios:

HH
TH
HT
TT

Obviously the TT possibility can be discounted because it does not result in one of the two being heads.

This lees us with three possibilities, only one of which has the other coin also being heads.

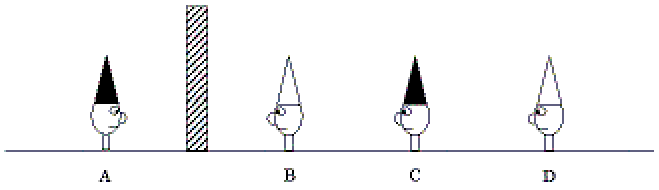
Therefore one third.

I think.

I usually get these wrong.

---

buried four fishermen up to their necks in the sand on the beach at low tide for keeping their fishing spot a secret from me. i put a hat on each of their heads and told them that one of them must shout out the correct color of their own hat or they will all be drowned by the incoming tide. i give them 10 minutes to do this. fisherman A and B can only see the sand dune i erected. fisherman C can see that fisherman B has a white hat on. fisherman D can see that C has a black hat, and B has a white hat. the fisherman have been told that there are four hats, two white and two black, so they know that they must have either a white or a black hat on. who shouts out the color of their hat and how do they know?



Fisherman C shouts out.

Fishermen A and B are in the same situation - they have no information to help them determine their hat colour so they can't answer. C and D realise this.

Fisherman D can see both B and C's hats. If B and C had the same colour hat then this would let D know that he must have the other colour.

When the time is nearly up, or maybe before, C realises that D isn't going to answer because he can't. C realises that his hat must be different to B's otherwise D would have answered. C therefore concludes that he has a black hat because he can see B's white one.

---

this is difficult to describe in words, so read this carefully, lest there be any confusion. you have a normal six sided cube. i give you six different colors that you can paint each side of the cube with (one color to each side). how many different cubes can you make?

different means that the cubes can not be rotated so that they look the same. this is important! if you give me two cubes and i can rotate them so that they appear identical in color, they are the same cube.

Let X be the number of "different" cubes (using the same definition as in the problem). Let Y be the number of ways you can "align" a given cube in space such that one face is pointed north, one is east, one is south, one is west, one is up, and one is down. (We're on the equator.) Then the total number of possibilities is X * Y. Each of these possibilities "looks" different, because if you could take a cube painted one way, and align it a certain way to make it look the same as a differently painted cube aligned a certain way, then those would not really be different cubes. Also note that if you start with an aligned cube and paint it however you want, you will always arrive at one of those X * Y possibilities.

How many ways can you paint a cube that is already "aligned" (as defined above)? You have six options for the north side, five options for the east side, etc. So the total number is 6! (that's six factorial, or 6 * 5 * 4 * 3 * 2 * 1). Note that each way you do it makes the cube "look" different (in the same way the word is used above). So 6! = X * Y.

How many ways can you align a given cube? Choose one face, and point it north; you have six options here. Now choose one to point east. There are only four sides that can point east, because the side opposite the one you chose to point north is already pointing south. There are no further options for alignment, so the total number of ways you can align the cube is 6 * 4.

Remember, Y is defined as the number of ways you can align the cube, so Y = 6 * 4. This gives us 6! = X * 6 * 4, so X = 5 * 3 * 2 * 1 = 30.

---

how many places are there on the earth that one could walk one mile south, then one mile east, then one mile north and end up in the same spot? to be precise, let's assume the earth is a solid smooth sphere, so oceans and mountains and other such things do not exist. you can start at any point on the sphere and walk in any direction you like.

think you've figured it out? i'll tell you now, there is more than one. in fact, there are more than two. also be advised that walking north from the north pole (or south from the south pole) is illogical and therefore does not enter into the problem. all normal assumptions about directions will be used.

there are **no** tricks involved with this question. it just forces you to really think about the problem to come up with all the solutions.

problem: how many places are there on the earth that one could walk one mile south, then one mile east, then one mile north and end up in the same spot? to be precise, let's assume the earth is a solid smooth sphere, so oceans and mountains and other such things do not exist. you can start at any point on the sphere and walk in any direction you like. think you've figured it out? i'll tell you now, there is more than one. in fact, there are more than two. also be advised that walking north from the north pole (or south from the south pole) is illogical and therefore does not enter into the problem. all normal assumptions about directions will be used.

there are no tricks involved with this question. it just forces you to really think about the problem to come up with all the solutions.

solution:

well the north pole is one such place.

then somewhere near the south pole such that when you walk one mile south you are at the point on the earth where the circumference is 1. that way when you walk 1 mile east, you end up back at the same point. and of course one mile north from there puts you back where you started. here is a drawing courtesy of jy. there may or may not be such a place in the northern hemisphere where walking a mile south puts you at the 1 mile circumference point on the earth.

i'm no geometry sphere expert, so someone will have to let me know if that is physically possible (i.e. i tend to think that if you walk n units south from any point on the northern part of a sphere, other than the north pole, it is impossible for the circumference to be n or less than n, but who knows?)

finally there are actually an infinite number of points. if we consider the case before where we went to the point with a circumference of 1, why not go to the point with a circumference of 1/2. then when you go a mile east, you loop around twice, and end up in the same spot. this holds true for 1/3, 1/4, 1/5, ... 1/n, etc.

---

there are three ants on a triangle, one at each corner. at a given moment in time, they all set off for a different corner at random. what is the probability that they don't collide?

Solved by Nahappan on Tuesday, April 10, 2001

solution: ants on a triangle

Consider the triangle ABC. We assume that the ants move towards different corners along the edges of the triangle.

Total no. of movements: 8 A->B, B->C, C->A A->B, B->A, C->A A->B, B->A, C->B A->B, B->C, C->B A->C, B->C, C->A A->C, B->A, C->A A->C, B->A, C->B A->C, B->C, C->B

Non-colliding movements: 2 A->B, B->C, C->A A->C, B->A, C->B

(i.e. the all ants move either in the clockwise or anti-clockwise direction at the same time)

P(not colliding) = 2/8 = 0.25

---

a. first i would just like to say i am so glad jenna got voted off survivor. yuck, i couldn't stand her ever since she accused that guy of eating beef jerky when he was chewing on grass.
b. recruit meyer on boot camp rocks. did you see how he made himself cry to get sympathy? and it worked! he is the funniest part of the show. i hope he wins.
c. i apologize for this site being **soooooo slow**. but that's why you get what you pay for. (i didn't pay anything so i shouldn't really expect anything in return.)
d. those screwy pirates are at it again. this time there are 13 pirates and they need to protect their treasure chest. they decide that they should only be able to open the chest if the majority (at least 7) agree that it should be opened. they ask a locksmith to come and put a specific number of locks on the safe. every lock must be opened to open the chest. there can be multiple keys for each lock, but each key only opens one lock (i.e. no skeleton keys). the locksmith can give more than one key to each pirate. how many locks should the locksmith use and what strategy should he use to distribute the keys, such that only when a majority of the pirates agree can the chest be opened?

problem: those screwy pirates are at it again. this time there are 13 pirates and they need to protect their treasure chest. they decide that they should only be able to open the chest if the majority (at least 7) agree that it should be opened. they ask a locksmith to come and put a specific number of locks on the safe. every lock must be opened to open the chest. there can be multiple keys for each lock, but each key only opens one lock (i.e. no skeleton keys). the locksmith can give more than one key to each pirate. how many locks should the locksmith use and what strategy should he use to distribute the keys, such that only when a majority of the pirates agree can the chest be opened?

---

a one armed surgeon with a hand wound needs to operate on three patients. the surgeon only has two gloves. how can he operate on the three patients in turn without risking exchange of fluids? (remember he only has one arm so he only needs to wear one glove at a time.)

problem: a one armed surgeon with a hand wound needs to operate on three patients. the surgeon only has two gloves. how can he operate on the three patients in turn without risking exchange of fluids? (remember he only has one arm so he only needs to wear one glove at a time.)

solution: the surgeon places both gloves on his hand (1 and 2). he operates on patient A. he then takes the top glove off (#2), leaving on the bottom glove (#1) and operates on patient B. then he carefully reverses glove #2, so the clean side is on the outside, and he places it on top of glove #1 which is on his hand, and operates on patient C.

this problem is kind of dumb because how's the surgeon going to change the gloves on his hand when he only has one hand. plus no offense, but how often do you come across a one-armed surgeon (i'm sure there are plenty of one-armed doctors, but a surgeon!?!). anyway, i had to make this problem child friendly and changing the story to the above was the only way to do it. consider for a minute what the initial problem was. the surgeon was just a guy, the patients were women, and the glove was... well, i won't insult your intelligence.

Thursday, March 15, 2001

gold chain aha:!!!

a man has a gold chain with 7 links. he needs the service of a laborer for 7 days at a fee of one gold link per day. however, each day of work needs to be paid for separately. in other words, the worker must be paid each day after working and if the laborer is ever overpaid he will quit with the extra money. also he will never allow himself to be owed a link.

what is the fewest # of cuts to the chain to facilitate this arrangement and how does that guarantee payment?

Can we get change back from the laborer?

If so, we cut one link to make a chain of 4 links, a chain of 2 links and the cut link itself.

Day 1, we give him the cut link
Day 2, we take back the cut link, give him the 2 link chain
Day 3, we give him the cut link
Day 4, we take back both the cut link and the 2 link chain, give him the 4 link chain
Day 5, we give him the cut link
Day 6, we take back the cut link, give him the 2 link chain
Day 7, we give him the cut link

---

this is a classic problem which i have heard many times before. this is the "harder" of the two problems, since in this one, you do not know if the invalid item weighs more or less than the others.

solving it is only half the battle. writing up a solution that anyone including your grandma could understand, is very hard.

problem: the evil king from before sends his own assassin to take care of the evil queen who tried to poison him. of course, her trusty guards catch the assassin before any harm is done. the queen notices that the assassin is quite handsome and doesn't really want to punish him by death. she decides to test his wisdom.

the queen gives the assassin 12 pills which are all completely identical in shape, smell, texture, size, except 1 pill has a different weight. the queen gives the man a balance and tells him that all the pills are deadly poison except for the pill of a different weight. the assassin can make three weighings and then must swallow the pill of his choice. if he lives, he will be sent back to the bad king's kingdom. if he dies, well, thats what you get for being an assassin.

only one pill is not poison and it is the pill which has a different weight. the assassin does not know if it weighs more or less than the other pills. how can he save his skin?

choose any eight of the pills and put four of them on each side of the balance.

there are two possibilities:

(1) one side of the balance comes out lighter. In this case, you know that the abnormal (safe) pill is one of the pills already on the balance. label the pills on the lighter side A B C and D, and the pills on the heavier side E F G and H. label the pills not on the balance NORM (you know they’re normal pills).

(2) the balance is even. in this case, you know that the abnormal (safe) pill is one of the pills not on the balance. label the pills already on the balance NORM, and label the four pills not on the balance I J K and L.

let’s proceed with possibility (1).

consider why the side ABCD came out higher than the side EFGH. this could be because:

A is the abnormal pill, and it’s lighter than the other pills.

B is the abnormal pill, and it’s lighter than the other pills.

C is the abnormal pill, and it’s lighter than the other pills.

D is the abnormal pill, and it’s lighter than the other pills.

E is the abnormal pill, and it’s heavier than the other pills.

F is the abnormal pill, and it’s heavier than the other pills.

G is the abnormal pill, and it’s heavier than the other pills.

H is the abnormal pill, and it’s heavier than the other pills.

now let’s make another weighing, with two of the ABCD pills on either side, and one of the EFGH pills on either side. for example, let’s weigh ABE versus CDF. how would this weighing come out given each of those 8 possibilities we just listed?

if A is the light pill, the ABE/CDF weighing will come out with ABE high.

if B is the light pill, the ABE/CDF weighing will come out with ABE high.

if C is the light pill, the ABE/CDF weighing will come out with ABE low.

if D is the light pill, the ABE/CDF weighing will come out with ABE low.

if E is the heavy pill, the ABE/CDF weighing will come out with ABE low.

if F is the heavy pill, the ABE/CDF weighing will come out with ABE high.

if G is the heavy pill, the ABE/CDF weighing will come out even.

if H is the heavy pill, the ABE/CDF weighing will come out even.

OK, so we observe how the ABE versus CDF weighing actually comes out.

(a) if it comes out even, then we know that the abnormal pill is either G or H. for our third weighing, we can weigh G against one of the pills we already know to be normal (one of the pills we labelled NORM). if it comes out even, then G is normal and H must be the abnormal pill. if it comes out uneven, then G is the abnormal pill.

(b) as we can see from our chart above, if the ABE/CDF weighing comes out with ABE high, then the situation is either: A is the light pill, B is the light pill, or F is the heavy pill.

(c) as we can see from our chart above, if the ABE/CDF weighing comes out with ABE low, then the situation is either: C is the light pill, D is the light pill, or E is heavy pill.

so in either situation (b) or (c), we have two possible light pills and one possible heavy pill. what we do in that case is we put one of the possible light pills and the possible heavy pill on one side of the scale, and two NORM pills on the other side of the scale. this is our third weighing. if it comes out even, then we know that the other possible light pill is the abnormal pill. if it comes out with the two NORM pills high, then we know that one of the pills on the other side is abnormally heavy, so we know that the possible heavy pill is the culprit. if it comes out with the two NORM pills low, then we know that one of the pills on the other side is abnormally light, so we know that the possible light pill on the scale is the culprit.

that takes care of case (1), where the first weighing came out uneven.

what about case (2), where the first weighing comes out even?

then we know the abnormal pill is one of I J K or L, and we have two weighings to find the abnormal pill in.

for our second weighing, we put I and J on one side of the scale, and two NORM pills on the other.

(a) if this comes out uneven, we know the abnormal pill is I or J; we weigh I against one NORM pill to see if I is abnormal and if it isn’t, we can conclude that J is the abnormal pill.

(b) if the IJ versus 2 NORM weighing comes out even, we know the abnormal pill is K or L; we weight K against one NORM pill to see if K is abnormal and if it isn’t, we can conclude that L is the abnormal pill.

finished.

C++ DEVELOPER QUESTIONS

(1) Expect to write code. If you're interviewing as a developer and they don't ask you to write code then they are not serious about finding out if you are qualified for the job. What are the odds the guy they hired right before you or the woman they hire right after you _is_ qualified. You'll have to work with them AND the manager who interviewed the lot of you.

(2) In that vein: Know how to manipulate C-style character strings. They are compact representations of data with a whole lot of library functions for manipulating them so they make good "on the fly" programming questions. They also involve pointers and null-termination. Gives the interviewee a chance to hang themselves. Several we like are atoi, itoa, and reverse a string in place. For more fun we can ask about multi-byte character sets (putting a character back into the input stream is the standard question here (okay, you start with getch and THEN have them put the character back).

(3) Pointers are fun. Know your basic data structures: linked list, binary tree, hash table. Know some of the pattern names in the GoF book and maybe even how to apply them (note: If you don't know what a design pattern is, read about them for your own personal improvement. They might help you get a job but more likely they will help you excel at whatever job you get while you're still digesting what a design pattern is).

(4) Expect some logic and estimation questions. How many cottonballs would it take to fill the jumbo jet you flew to our interview in? How many movie theatres are there in London, Ontario? With these types of questions gather information before charging ahead and be sure to state your assumptions. If your assumptions are wrong the interviewer will guide you to the question they want answered. If you don't state your assumptions then they can't help you and can only grade what you say/write. Ask the population of London, Ontario or state an assumption about it. Make a guess about the size of a cottonball and the dimensions of the plane you flew on.

(5) Know simple area and perimeter formulae (square, triangle, circle, sphere, cube, pyarmid, etc.).

(6) When writing code: Write comments. State, in writing, what your assumptions and expectations are. Don't go overboard but a single sentence explaining what the limitations of your atoi function are will endear you to the technical members of the interview team.

Hope this helps and good luck.
-bcl

Comment from Axter
Date: 12/18/2003 09:24PM PST

Comme
nt

>>Expect to write code. If you're interviewing as a developer and they don't ask you to write code then they are not serious about finding out if you are qualified for the job. What are the odds the guy they hired right
>>before you or the woman they hire right after you _is_ qualified. You'll have to work with them AND the manager who interviewed the lot of you.

I have had very few interviews in which they ask for on the spot code.

>>multi-byte character sets (putting a character back into the input stream is the standard question here (okay, you start with getch and THEN have them put the character back).

In the states, very few interviewer will touch on multi-byte characters, unless it's used in their business.

Number (3) is very important. (Good Stuff!)

Describe the three main parts of OO.

1. polymorphism
2. encapsulation
3. inheritance

What is an abstract class?

1. A class that has at least one pure virtual function
2. Can not be instantiated

Why use virtual destructors?

Abstract class should always have a virtual destructor, unless absolutely sure that the descendant class will not be deleted via base class pointer.
This is true even if the descendant class has no objects to destroy. Otherwise according to the C++ standard, it will result in undefined behavior.
Section 5.3.5 in the Standard:
"3. In the first alternative (delete object), if the static type of the operand is different from its dynamic type, the static type shall be a base class of the operand's dynamic type and the static type shall have a virtual destructor or the behavior is undefined."

What is the difference between putting the initialization of an object in the initialization part of the constructor versus in the body of the constructor?

What is the difference between creating an object using new and creating an object using malloc?

New calls the object’s constructor and malloc does not.
Any object created with new must be freed using delete.

What’s the difference between a copy constructor and an assignment operator?

A copy constructor initializes un-initialized variables, where-as an assignment operator must deal with well constructed objects.
An assignment operator should protect itself against self assignment. (check if equal to this)
A copy constructor can copy an object that has constant types and/or reference types. In general, an assignment operator can not correctly copy an object that has reference and/or constant types.

What is coupling?

Coupling is the degree to which software components depend on each other.

What is abstract coupling?

Given a class X that maintains a reference to an abstract class Y, class X is said to be abstractly coupled to Y. It's called abstract coupling because X refers to a type of object, and not a concrete object.

What is "cohesion and coupling"?

Have any new keywords been added to C++? What about types?

1. export
2. mutable
3. types (wchar_t)

What is the difference between a pointer and a reference variable?

1. A reference must be initialized to point to an object.
2. A reference can not change it's pointer
3. Pointer arithmetic can not be performed on a reference

When should a function be declared with a pointer argument over a reference argument?

1. When the argument needs to change what it's pointing to.
2. When the object needs to be deleted or created

Explain what is the difference between dynamic_cast to a pointer vs. dynamic_cast to a reference.

What possible things can happen when operator new fails to allocate memory?

Unless an allocation function is declared with an empty exceptionspecification (15.4), throw(), it indicates failure to allocate storage by throwing a bad_alloc exception (clause 15, 18.4.2.1); it returns a non-null pointer otherwise.

What is polymorphism?

1. 'Polymorphism is the capability of different objects to react in an individual manner to the same message.'
2. Polymorphism is the object-oriented buzz-word for dynamic binding, which itself is an OO buzz-word
3. Polymorphism refers to the capability of an operation to operate on different entities and to exhibit behavior appropriate for the entity operated on. This is not directly related to object oriented programming although object oriented programming supports a certain kind of polymorphism, namely dynamic polymorphism or late binding: An operation accepts objects and sends messages to them (in C++ these are virtual functions, in other language this may be realized differently). The objects respond to the messages according to their specific definition, in C++ according to their class. There are other kinds of polymorphism, eg. static polymorphism which is resolved at compile time: The operation just uses functions on its arguments (these may be methods in C++ but it can be just functions as well) which are defined to do the right thing for the arguments they get passed. In C++ this kind of polymorphism is realized as templates and there is no direct counterpart in object oriented programming. Another kind of polymorphism, neither supported by C++ nor by object oriented programming is the use of higher order functions in functional programming languages: The higher order function operates on other function to achieve a result depending on the functions it gets passed.
4. polymorphism (pol`e-mor'fiz-?m) noun: In an object-oriented programming language, the ability to redefine a routine in a derived class (a class that inherited its data structures and routines from another class). Polymorphism allows the programmer to define a base class that includes routines that perform standard operations on groups of related objects, without regard to the exact type of each object. The programmer then redefines the routines in the derived class for each type, taking into account the characteristics of the object.
5. Parmetric polymorphism is exclusively a compile-time feature.

Why multiple inheritance?

What are mixins?

A class that provides some, but not all of the implementation for a virtual base class is often called a mixin.

What is an adapter?

A component that modifies the interface of another component is called an adaptor.
A usage example would be using an adapter to get std::string to work with code that access MFC-CString functions. An adaptor can be used to get the constant char pointer.

What is a bridge?

The Bridge pattern isn't just for establishing/changing implementation at run-time, it can also be used to help reduce the total number of classes necessary in the hierarchy. For example if we had 3 classes derived from IA and 3 derived from A, then we can combine them to form 9 differently behaved objects. Using multiple inheritance, we avoid the Bridge pattern, but end up needing 9 sub-classes...

What is a Pimpl design?

A Pimpl allows for hiding an objects declaration and implementation behind a forward declared class pointer. It's used to reduce compile time dependencies.

What is layering?

It's the process of building one class on top of another class by having the layering class contain an object of the layered class as a data member. Layering is also known as composition, containment, and embedding.

What might I look for to consider an association between objects as an aggregation?

The lifetime of the aggregate object and its owner are identical. An aggregate object avoids exposure of its implementation.

What are the different forms of aggregation?

How are the lifetimes of containers and contained objects related in the various forms of aggregation?

The lifetime of the aggregate object and its owner are identical

What does "mutable" mean?

An object that can be modified inside a constant type member function.

Explain the Liskov Substition Principle.

The is-a rule for public base class.

What is a virtual base class?

A virtual base class is a class that has one or more virtual functions.

How is a pure virtual member function defined?

Make the function equaled to zero in the class declaration, and use the virtual keyword.

What is bit slicing or object slicing?

When a derived class has parts sliced off. This occurs when a derived class is passed by value via the base type.

What is the dominance rule?

For virtual functions the dominance rule is what guarantees that the same function is called independently of the static type of the pointer, reference, or name of the object for which it is called

Suppose I have a class B that derives from class A and I wanted to reuse some, but not all of the methods of B. How would I restrict access to the superclass' methods selectively?

Differentiate between subtyping and subclassing.

- a class = implementation module
- a type = abstract data type = behavior as given by some specification
- subclass inherits **implementation**
- subtype inherits specification (behavior). (**interface**)

How can subtyping and subclassing be approximated or implemented in C++?

1. subtyping can be done via pure virtual functions
2. subclassing can be done with non virtual functions

Are you familiar with any design methodology? Booch? OMT? Use cases?

What is an Abstract Factory and why would I want to use one?

An interface that allows for the creation of objects without specifying the concrete class.

Questions on STL and iostreams.

... and a whole lot more!
I tend to look for someone who not only has an intricate technical grasp of the language, but has strong skills in OOA/OOD.
IMHO, analysis and design skills are more important as they are language independent, however, applying these skills to practical systems are as important.

Wanted: Senior C++ Programmer

by Al Stevens

Good morning, Mr. Phelps. The title of this month's column is a typical entry taken from the help-wanted section of the local classified ads. Does it get your attention? Add your address and phone number to the ad, publish it in your local paper, and, depending on where you live, you will get lots of calls and mail. Headhunters will deluge you with résumés, every one of which has the token C++ somewhere near the top of the first page. Programmers who want to reap the harvest of the C++ demand will polish up their résumés to emphasize their C++ exposure and send them in.

Your mission, should you choose to accept it, is to find the best applicant, the one capable of filling the senior C++ programming position, from among the many.

Last month, I listed some questions that you can ask people who apply for a C++ programming job. This month I'll discuss what I think are good answers to those questions. I'm writing this column in June, and the August issue is not on the street yet, so, although I asked for your comments, you haven't had that opportunity. Once again, this technique is my own device, is influenced by my opinions, and needs polish. I'm eager to hear all comments and criticisms.

The questions are in three categories:

* The first category probes whether applicants have a rudimentary understanding of the differences between C and C++.

* The second category examines their grasp of class design.

* The third determines how well they have kept abreast of language changes that the ANSI committee has proposed and approved.

I developed this technique after watching a client conduct interviews of many applicants for a single position. It became obvious that there was no structure to the art of the interview. The interviewer understood the problems of the project, but she is not a C++ programmer. Consequently, her technical questions were superficial ("Have you ever programmed with OWL?", for example) and the answers revealed nothing important about the applicant's abilities. When she asked me to help, I realized after a couple of interviews that my extemporaneous style needed better organization. I needed a plan. This set of questions is the result.

The first question you should ask is if the applicant is a devoted reader of this column. If so, you're on your own. The programmer has probably read and memorized these answers by now. Hire 'em. Readers of this column are always a good bet, I bet.

After that, the questions you ask are determined by how applicants represent their knowledge and experience. Anyone who really wants the job tries to match aspects of their experience with an interviewer's questions. It's a natural tendency.

There is an art to résumé writing, too, that emphasizes work experiences to fit the job requirements no matter how trivial the experience was. Some headhunters are really good at it. Therefore, don't depend on the details of a résumé to properly identify an applicant's abilities. But you do need to make that determination because if you ask too many questions that are beyond the scope of a programmer's knowledge, the interview deteriorates into a confrontation, and the applicant stops wanting to work for you. Consequently, the first three questions are designed to reveal how applicants want their skills to be regarded and should permit you to follow with the appropriate questions chosen from the three groups.

Questions to Qualify an Applicant

Q: Do you have a basic understanding of C and C++ and their similarities and differences?

A: Yes.

If the answer is "yes," proceed and plan to ask all the questions in the first group. If not, the interview for a C++ programmer's job should be over, and you should be talking about other employment opportunities.

Q: Have you participated in the design of C++ classes to support an application's problem domain?

A: Yes or No.

If the answer is "yes," plan to include the questions in the second group. Applicants usually want to describe the details of the systems they have designed. Pay attention if you understand the nature of the application. If not, pretend to pay attention. If they have not actually done any class design, ask if they think they understand it. If so, include the second set of questions.

Q: Do you keep up with what the ANSI C++ Standards committee is doing?

A: Yes or No.

Most people do not. Few people have the time. But occasionally there is that rare soul who reads all the magazines and books, owns a copy of the draft standard, and regularly tracks the C++ news groups on the net. If an applicant claims to be one of them, include the third group of questions in your interrogation.

The Art of the Interview

With your applicants qualified as to how they represent themselves, ask the questions from the chosen three groups. Mix the questions and keep a light but detached attitude going if you can. As soon as you get into details, many people tighten up. They know they are being tested, and as they try to interpret the question to figure out what you want to hear, sometimes they lose sight of the objective, which is to expose what they know rather than what their intuition tells them is an acceptable answer.

Maintain your detachment. Be cordial, but don't get chummy just yet. Remember, if you hire this person to work with you on your typically understaffed, under- budget, overdue C++ programming project, the two of you are going to spend a lot of tense moments together. Observe how the applicant handles the pressure of the interview. If your shop is typical, the two of you will have to routinely deal with a lot more pressure than this.

Don't expect everyone to get everything right (unless they read this column, of course, and fibbed about the first question).

Questions for All Applicants

Here's the first group of questions and my opinions about what some acceptable answers would be. These questions do not cover C++ wall-to-wall, of course. I selected them as typical of the kind of knowledge that all C++ programmers should be expected to possess. There are five questions. Three correct answers is a good score.

Q: How do you link a C++ program to C functions?

A: By using the extern "C" linkage specification around the C function declarations.

Programmers should know about mangled function names and type-safe linkages. Then they should explain how the *extern* "*C*" linkage specification statement turns that feature off during compilation so that the linker properly links function calls to C functions. Another acceptable answer is "I don't know. We never had to do that." Merely describing what a linker does indicates that the programmer does not understand the issue that underlies the question.

Q: Explain the scope resolution operator.

A: It permits a program to reference an identifier in the global scope that has been hidden by another identifier with the same name in the local scope.

The answer can get complicated. However, it should start with "::". If the programmer is well into the design or use of classes that employ inheritance you might hear a lot about overriding member function overrides to explicitly call a function higher in the hierarchy. That's good to know, but ask specifically about global scope resolution. You're looking for a description of C++'s ability to override the particular C behavior where identifiers in the global scope are always hidden by like identifiers in a local scope.

Q: What are the differences between a C++ *struct* and C++ *class*?

A: The default member and base-class access specifiers are different.

This is one of the commonly misunderstood aspects of C++. Believe it or not, many programmers think that a C++ *struct* is just like a C *struct*, while a C++ class has inheritance, access specifiers, member functions, overloaded operators, and so on. Some of them have even written books about C++. Actually, the C++ *struct* has all the features of the *class*. The only differences are that a *struct* defaults to public member access and public base-class inheritance, and a *class* defaults to the private access specifier and private base-class inheritance. Getting this question wrong does not necessarily disqualify an applicant. Getting it right is a definite plus.

Saying, "I don't know" is definitely the wrong answer. I advance an unusual position about this. C++ programmers should at least believe that they know the differences, even when they are wrong about them. Getting it wrong is, therefore, right. You can explain the true difference in the interview and advance the programmer's knowledge. If they disagree vociferously, you have an opportunity to observe how they handle contentious debate when they are wrong and don't know it yet.

Q: How many ways are there to initialize an *int* with a constant?

A: Two.

There are two formats for initializers in C++ as shown in the example that follows. The first format uses the traditional C notation. The second format uses constructor notation.

int foo = 123;

int bar (123);

It's acceptable when a programmer does not know about the second notation, although they should certainly know about the first one. Many old-timer C programmers who made the switch to C++ never use the second idiom, although some wise heads of C++ profess to prefer it. If your applicant is quick with the right answer, that's a good sign.

Q: How does throwing and catching exceptions differ from using *setjmp* and *longjmp*?

A: The throw operation calls the destructors for automatic objects instantiated since entry to the *try* block.

Exceptions are in the mainstream of C++ now, so most programmers, if they are familiar with *setjmp* and *longjmp*, should know the difference. Both idioms return a program from the nested depths of multiple function calls to a defined position higher in the program. The program stack is "unwound" so that the state of the program, with respect to function calls and pushed arguments, is restored as if the calls had not been made. C++ exception handling adds to that behavior the orderly calls to the destructors of automatic objects that were instantiated as the program proceeded from within the *try* block toward where the *throw* expression is evaluated.

Applicants might think you want to hear about the notational differences between the two idioms. Let them proceed to explain the syntax of *try* blocks, *catch* exception handlers, and *throw* expressions. Then ask them specifically what happens in a *throw* that does not happen in a *longjmp*. Their answer should reflect an understanding of the behavior described in the previous answer.

One valid reason for not knowing about exception handling is that the applicant's experience is exclusively with older C++ compilers that do not implement exception handling. I would prefer that they have at least heard of exception handling, though. Another marginally acceptable reason is that their former supervisors and designers did not mandate and specify the use of exception handling in programs. In that case get the names of those supervisors and designers so that you can decline their applications if they should come a'knocking.

It is not unusual for C and C++ programmers to be unfamiliar with *setjmp/ longjmp*. Those constructs are not particularly intuitive. A C programmer who has written recursive descent parsing algorithms will certainly be familiar with *setjmp/longjmp*. Others might not, and that's acceptable. In that case, they won't be able to discuss how *setjmp/longjmp* differs from C++ exception handling, but let the interview turn into a discussion of C++ exception handling in general. That conversation will reveal a lot about a programmer's understanding of C++.

Questions for Class Designers

The second group of questions explores the applicant's knowledge of class design. There are eight questions. Five out of eight is a good score.

Q: What is your reaction to this line of code?

delete this;

A: It's not a good practice.

Many applicants will look at you like you are nuts. They've never heard of this usage, and it's never occurred to them. That's a very good answer. Perhaps they will try to explain the behavior of the statement. Ask them to contemplate its consequences. Two quite acceptable reactions are, "Don't do it," and "Don't do it unless you really know what you are doing and you are a masochist."

A good programmer will insist that you should absolutely never use the statement if the class is to be used by other programmers and instantiated as static, extern, or automatic objects. That much should be obvious.

The code has two built-in pitfalls. First, if it executes in a member function for an extern, static, or automatic object, the program will probably crash as soon as the *delete* statement executes. There is no portable way for an object to tell that it was instantiated on the heap, so the class cannot assert that its object is properly instantiated. Second, when an object commits suicide this way, the using program might not know about its demise. As far as the instantiating program is concerned, the object remains in scope and continues to exist even though the object did itself in. Subsequent dereferencing of the pointer can and usually does lead to disaster. I think that the language rules should disallow the idiom, but that's another matter.

In *More Effective C++* (Addison-Wesley, 1996), Scott Meyers devotes one of his items to "*delete this*," implying that there are valid applications for the idiom and advancing contrived code kludges to make it seem to work better. A programmer who has read this otherwise very good book might think that the practice is acceptable. Experience leads me to disagree.

Q: What is a default constructor?

A: A constructor that has no arguments.

If you don't code one, the compiler provides one if there are no other constructors. If you are going to instantiate an array of objects of the class, the class must have a default constructor.

Q: What is a conversion constructor?

A: A constructor that accepts one argument of a different type.

The compiler uses this idiom as one way to infer conversion rules for your class. A constructor with more than one argument and with default argument values can be interpreted by the compiler as a conversion constructor when the compiler is looking for an object of your constructor's type and sees an object of the type of the constructor's first argument.

Q: What is the difference between a copy constructor and an overloaded assignment operator?

A: A copy constructor constructs a new object by using the content of the argument object. An overloaded assignment operator assigns the contents of an existing object to another existing object of the same class.

First, the applicant must know that a copy constructor is one that has only one argument of the same type as the constructor. The compiler invokes a copy constructor wherever it needs to make a copy of the object, for example to pass an argument by value. If you do not provide a copy constructor, the compiler creates a member- by-member copy constructor for you.

You can write overloaded assignment operators that take arguments of other classes, but that behavior is usually implemented with implicit conversion constructors. If you do not provide an overloaded assignment operator for the class, the compiler creates a default member- by-member assignment operator.

This discussion is a good place to get into why classes need copy constructors and overloaded assignment operators. If the applicant discusses these with respect to data member pointers that point to dynamically allocated resources, the applicant probably has a good grasp of the problem.

Q: When should you use multiple inheritance?

A: There are three acceptable answers: "Never," "Rarely," and "When the problem domain cannot be accurately modeled any other way."

There are some famous C++ pundits and luminaries who disagree with that third answer, but I will accept it.

Let's digress to consider this issue lest your interview turn into a religious debate. Consider an *Asset* class, *Building* class, *Vehicle* class, and *CompanyCar* class. All company cars are vehicles. Some company cars are assets because the organizations own them. Others might be leased. Not all assets are vehicles. Money accounts are assets. Real estate holdings are assets. Some real estate holdings are buildings. Not all buildings are assets. Ad infinitum. When you diagram these relationships, it becomes apparent that multiple inheritance is a likely and intuitive way to model this common problem domain. The applicant should understand, however, that multiple inheritance, like a chainsaw, is a useful tool that has its perils, needs respect, and is best avoided except when nothing else will do.

Q: What is a virtual destructor?

A: The simple answer is that a virtual destructor is one that is declared with the virtual attribute.

The behavior of a virtual destructor is what is important. If you destroy an object through a pointer or reference to a base class, and the base-class destructor is not virtual, the derived-class destructors are not executed, and the destruction might not be complete.

Q: Explain the ISA and HASA class relationships. How would you implement each in a class design?

A: A specialized class "is" a specialization of another class and, therefore, has the ISA relationship with the other class. An *Employee ISA Person*. This relationship is best implemented with inheritance. Employee is derived from *Person*. A class may have an instance of another class. For example, an employee "has" a salary, therefore the *Employee* class has the HASA relationship with the *Salary* class. This relationship is best implemented by embedding an object of the *Salary* class in the *Employee* class.

The answer to this question reveals whether the applicant has an understanding of the fundamentals of object- oriented design, which is important to reliable class design.

There are other relationships. The USESA relationship is when one class uses the services of another. The *Employee* class uses an object (*cout*) of the *ostream* class to display the employee's name on the screen, for example. But if the applicant gets ISA and HASA right, you don't need to go any further.

Q: When is a template a better solution than a base class?

A: When you are designing a generic class to contain or otherwise manage objects of other types, when the format and behavior of those other types are unimportant to their containment or management, and particularly when those other types are unknown (thus, the genericity) to the designer of the container or manager class.

Prior to templates, you had to use inheritance; your design might include a generic *List* container class and an application-specific *Employee* class. To put employees in a list, a *ListedEmployee* class is multiply derived (contrived) from the *Employee* and *List* classes. These solutions were unwieldy and error-prone. Templates solved that problem.

Questions for ANSI-Knowledgeable Applicants

There are six questions for those who profess knowledge of the progress of the ANSI committee. If you claim to have that much interest in the language, you should know the answers to all these questions.

Q: What is a mutable member?

A: One that can be modified by the class even when the object of the class or the member function doing the modification is *const*.

Understanding this requirement implies an understanding of C++ *const*, which many programmers do not have. I have seen large class designs that do not employ the *const* qualifier anywhere. Some of those designs are my own early C++ efforts. One author suggests that some programmers find *const* to be such a bother that it is easier to ignore *const* than to try to use it meaningfully. No wonder many programmers don't understand the power and implications of *const*. Someone who claims to have enough interest in the language and its evolution to keep pace with the ANSI deliberations should not be ignorant of *const*, however.

Q: What is an *explicit* constructor?

A: A conversion constructor declared with the explicit keyword. The compiler does not use an *explicit* constructor to implement an implied conversion of types. It's purpose is reserved explicitly for construction.

Q: What is the Standard Template Library?

A: A library of container templates approved by the ANSI committee for inclusion in the standard C++ specification.

A programmer who then launches into a discussion of the generic programming model, iterators, allocators, algorithms, and such, has a higher than average understanding of the new technology that STL brings to C++ programming.

Q: Describe run-time type identification.

A: The ability to determine at run time the type of an object by using the *typeid* operator or the *dynamic_cast* operator.

Q: What problem does the namespace feature solve?

A: Multiple providers of libraries might use common global identifiers causing a name collision when an application tries to link with two or more such libraries. The namespace feature surrounds a library's external declarations with a unique namespace that eliminates the potential for those collisions.

This solution assumes that two library vendors don't use the same namespace identifier, of course.

Q: Are there any new intrinsic (built-in) data types?

A: Yes. The ANSI committee added the *bool* intrinsic type and its *true* and *false* value keywords.

Other apparent new types (string, complex, and so on) are implemented as classes in the Standard C++ Library rather than as intrinsic types.

What Not to Do

Don't ask the applicant to write a program on the spot. Many programmers, myself included, like to ponder a program for a while before writing any code. We would consider a one-hour deadline as a threat. We would be unwilling to let anyone see such a program. You'll get a far better understanding of the applicant's programming skills by asking a former employer about the programmer's ability to write reliable, extensible, maintainable, and timely code that fulfills the user's requirements.

Don't turn these questions into a written test. Some people don't take tests well, particularly when the answers involve narrative text. Many fine programmers are not articulate writers. Others simply hate tests and shut down mentally.

Case in point. My pal Bill Chaney was the highest-time student pilot I ever met. Over a period of several years, he accumulated close to 1000 hours on his student pilot's license. I would have trusted Bill to fly one of my children anywhere in that old Cessna 150 of his, but he was intimidated by the environment of the written examination and could not pass it. There's a happy ending: Several of us ganged up on Bill, crammed him on the written exam, saturated him with a positive mental attitude, and railroaded him through passage of the test. Bill is a legal pilot today.

There's another reason to avoid the written test, one that is driven by social changes in our culture. Last month I said, "For some reason, the Labor Department doesn't want you disqualifying anyone based on their inability to demonstrate that they can do the job." Now I'll expand on that statement, and perhaps stir a bit of controversy.

During my government client's quest for employees, she was questioned by the local EEO representative about her testing practices. They were particularly concerned about whether she was administering, and specifically, grading tests. There is a strong belief that certain segments of our society, like my pal Bill Chaney, do not take tests well, that those segments can be identified by ethnic divisions, and that testing them provides the potential for discrimination. That belief, or the fear of those who hold it, drives official policy.

The EEO is not consistent in this concern. You can't carry passengers in a 747 without proving your experience and ability and passing several of the government's oral and written graded tests. You don't get anywhere near my gall bladder, retirement account, or household wiring without a certificate on the wall proving that you have scored passing grades on a bevy of written tests. The potential to discriminate is overlooked when something important like our lives or our money is at stake.

My parents' generation was the last one that discriminated against ethnic and gender boundaries with the sanction of law to back them up. (My parents didn't do it, but their generation did and the law allowed it.) Many of my generation inherited and retained that bias even when the sanctions were removed through legislation and judicial decree. Most of the newer generation is free of that bias, I hope, at least intellectually if not emotionally. But the swing of the pendulum is not yet dead center. If you administer written tests to programmers, grade those tests, and use those grades to disqualify someone for employment, you might find yourself the object of a discrimination investigation.

So, ask those questions and take notes. But don't write grades on a score sheet. Or, in case you do, there is a device in our office named "Hillary." Get one. (It's also called a shredder.) As usual, if you or any of your interviewers are killed or captured, *DDJ* will disavow any knowledge of your activities.

Interviewing at Microsoft used to be different from Interviwing at other companines. However, with the advent of a new economy and the Internet things have changed dramatically in the past few years.

More and more companies are adopting Microsoft Interviewing philosophy and hence their questions. Here is a typical set of Interview questions asked for an entry level Software Design Engineer(SDE) or Internship position.

- If you had an infinite supply of water and a 5 quart and 3 quart pail, how would you measure exactly 4 quarts?
- If you are on a boat and you throw out a suitcase, will the level of water increase?
- On an average, how many times would you have to open the Seattle phone book to find a specific name?
- 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?
- If you look at a clock and the time is 3:15, what is the angle between the hour and the minute hands? (The answer to this is not zero!)
- What new feature would you add to MSWORD if you were hired?
- Why did you pick the school you graduated from?
- Why do you want to work for Microsoft?
- How many Gas stations are there in the US?
- How would you weigh a plane without using scales?
- How would you move Mt. Everest?
- Two MIT math graduates bump into each other at Fairway on the upper west side. They hadn't seen each other in over 20 years.
  The first grad says to the second: "how have you been?"
  Second: "Great! I got married and I have three daughters now"
  First: "Really? how old are they?"
  Second: "Well, the product of their ages is 72, and the sum of their ages is the same as the number on that building over there.."
  First: "Right, ok.. oh wait.. hmmmm.., I still don't know"
  second: "Oh sorry, the oldest one just started to play the piano"
  First: "Wonderful! my oldest is the same age!"

  Problem: How old are the daughters?

- Why are beer cans tapered at the top and bottom?
- Why is it that hot water in a hotel comes out instantly but at home it takes time?
- How many times a day a clock's hands overlap?

- Mike has $20 more than Todd. How much does each have given that combined they have $21 between them. You can't use fractions in the answer.(Hint: This is a trick question, pay close attention to the condition)
- There are four dogs, each at the counter of a large square. Each of the dogs begins chasing the dog clockwise from it. All of the dogs run at the same speed. All continously adjust their direction so that they are always heading straight towards their clockwise neighbor. How long does it take for the dogs to catch each other? Where does this happen? (Hint: Dog's are moving in a symmetrical fashion, not along the edges of the square).

C++ based applications---

C/C++ Questions

These are sample questions. Personally I hate asking questions related to code. If you want some old question papers of some Indian IT industries , You many find it here

1. What is the output of **printf("%d")**
2. What will happen if I say **delete this**
3. Difference between "C structure" and "C++ structure".
4. Diffrence between a "assignment operator" and a "copy constructor"
5. What is the difference between "overloading" and "overridding"?
6. Explain the need for "Virtual Destructor".
7. Can we have "Virtual Constructors"?
8. What are the different types of polymorphism?
9. What are Virtual Functions? How to implement virtual functions in "C"
10. What are the different types of Storage classes?
11. What is Namespace?
12. What are the types of STL containers?.
13. Difference between "vector" and "array"?
14. How to write a program such that it will delete itself after exectution?
15. Can we generate a C++ source code from the binary file?
16. What are inline functions?
17. Talk sometiming about profiling?
18. How many lines of code you have written for a single program?
19. What is "strstream" ?
20. How to write Multithreaded applications using C++?
21. Explain "passing by value", "passing by pointer" and "passing by reference"
22. Write any small program that will compile in "C" but not in "C++"
23. Have you heard of "mutable" keyword?
24. What is a "RTTI"?
25. Is there something that I can do in C and not in C++?
26. Why preincrement operator is faster than postincrement?
27. What is the difference between "calloc" and "malloc"?
28. What will happen if I allocate memory using "new" and free it using "free" or allocate sing "calloc" and free it using "delete"?
29. What is Memory Alignment?
30. Explain working of printf.

31. Difference between "printf" and "sprintf".
32. What is "map" in STL?
33. When shall I use Multiple Inheritance?
34. What are the techniques you use for debugging?
35. How to reduce a final size of executable?
36. Give 2 examples of a code optimization.

Java Interview Questions

1. Meaning - Abstract classes, abstract methods
2. Difference - Java,C++
3. Difference between == and equals method
4. Explain Java security model
5. Explain working of Java Virtual Machine (JVM)
6. Difference : Java Beans, Servlets
7. Difference : AWT, Swing
8. Disadvantages of Java
9. What is BYTE Code ?
10. What gives java it's "write once and run anywhere" nature?
11. Does Java have "goto"?
12. What is the meaning of "final" keyword?
13. Can I create final executable from Java?
14. Explain Garbage collection mechanism in Java
15. Why Java is not 100% pure object oriented language?
16. What are interfaces? or How to support multiple inhertance in Java?
17. How to use C++ code in Java Program?
18. Difference between "APPLET" and "APPLICATION"

Visual Basic Interview Questions

1. 3 main differences between flexgrid control and dbgrid control
2. ActiveX and Types of ActiveX Components in VB
3. Advantage of ActiveX Dll over Active Exe
4. Advantages of disconnected recordsets
5. Benefit of wrapping database calls into MTS transactions
6. Benefits of using MTS
7. Can database schema be changed with DAO, RDO or ADO?
8. Can you create a tabletype of recordset in Jet - connected ODBC database engine?
9. Constructors and distructors
10. Controls which do not have events
11. Default property of datacontrol
12. Define the scope of Public, Private, Friend procedures?
13. Describe Database Connection pooling relative to MTS
14. Describe: In of Process vs. Out of Process component. Which is faster?

15. Difference between a function and a subroutine, Dynaset and Snapshot,early and late binding, image and picture controls,Linked Object and Embedded Object,listbox and combo box,Listindex and Tab index,modal and moduless window, Object and Class,Query unload and unload in form, Declaration and Instantiation an object?
16. Draw and explain Sequence Modal of DAO
17. How can objects on different threads communicate with one another?
18. How can you force new objects to be created on new threads?
19. How does a DCOM component know where to instantiate itself?
20. How to register a component?
21. How to set a shortcut key for label?
22. Kind of components can be used as DCOM servers
23. Name of the control used to call a windows application
24. Name the four different cursor and locking types in ADO and describe them briefly
25. Need of zorder method, no of controls in form, Property used to add a menus at runtime, Property used to count number of items in a combobox,resize a label control according to your caption.
26. Return value of callback function, The need of tabindex property
27. Thread pool and management of threads within a thread pool
28. To set the command button for ESC, Which property needs to be changed?
29. Type Library and what is it's purpose?
30. Types of system controls, container objects, combo box
31. Under the ADO Command Object, what collection is responsible for input to stored procedures?
32. VB and Object Oriented Programming
33. What are the ADO objects? Explain them.
34. What are the different compatibility types when we create a COM component?
35. What do ByVal and ByRef mean and which is the default?
36. What does Option Explicit refer to?
37. What does the Implements statement do?
38. What is OLE and DDE? Explain.
39. What is the difference between Msgbox Statement and MsgboxQ function?
40. What keyword is associated with raising system level events in VB?
41. What methods are called from the ObjectContext object to inform MTS that the transaction was successful or unsuccessful?
42. What types of data access have you used.
43. What was introduced to Visual Basic to allow the use of Callback Functions?
44. Which controls can not be placed in MDI?
45. Which controls have refresh method, clear method
46. Which Property is used to compress a image in image control?
47. Which property of menu cannot be set at run time?
48. Which property of textbox cannot be changed at runtime and What's the maximum size of a textbox?
49. Which tool is used to configure the port range and protocols for DCOM communications?

C++ Questions

- What are the major differences between C and C++?
  - What are the differences between `new` and `malloc`?
  - What is the difference between `delete` and `delete[]`?
  - What are the differences between a struct in C and in C++?
  - What are the advantages/disadvantages of using `#define`?
  - What are the advantages/disadvantages of using `inline` and `const`?
- What is the difference between a pointer and a reference?
  - When would you use a pointer? A reference?
  - What does it mean to take the address of a reference?
- What does it mean to declare a function or variable as `static`?
- What is the order of initalization for data?
- What is name mangling/name decoration?
  - What kind of problems does name mangling cause?
  - How do you work around them?
- What is a class?
  - What are the differences between a struct and a class in C++?
  - What is the difference between public, private, and protected access?
  - For `class CFoo { };` what default methods will the compiler generate for you>?
  - How can you force the compiler to not generate them?
  - What is the purpose of a constructor? Destructor?
  - What is a constructor initializer list?
  - When *must* you use a constructor initializer list?
  - What is a:
    - Constructor?
    - Destructor?
    - Default constructor?
    - Copy constructor?
    - Conversion constructor?
  - What does it mean to declare a...
    - member function as `virtual`?
    - member function as `static`?
    - member function as `static`?
    - member variable as `static`?
    - destructor as `static`?
  - Can you explain the term "resource acquisition is initialization?"
  - What is a "pure virtual" member function?
  - What is the difference between public, private, and protected inheritance?
  - What is virtual inheritance?
  - What is placement `new`?
  - What is the difference between `operator new` and the `new` operator?
- What is exception handling?
  - Explain what happens when an exception is thrown in C++.

    - What happens if an exception is not caught?
    - What happens if an exception is throws from an object's constructor?
    - What happens if an exception is throws from an object's destructor?
    - What are the costs and benefits of using exceptions?
    - When would you choose to return an error code rather than throw an exception?
- What is a template?
- What is partial specialization or template specialization?
- How can you force instantiation of a template?
- What is an iterator?
- What is an algorithm (in terms of the STL/C++ standard library)?
- What is `std::auto_ptr`?
- What is wrong with this statement? `std::auto_ptr ptr(new char[10]);`
- It is possible to build a C++ compiler on top of a C compiler. How would you do this? (Note: I've only asked this question once; and yes, he understood that I was asking him how cfront was put together. We hired him.)

Code Snippets

I like to put these up on a whiteboard, and ask questions about them.

What output does the following code generate? Why?

What output does it generate if you make A::Foo() a pure virtual function?

```
#include <iostream>

using namespace std;

 class A {
public:
        A() {
                this->Foo();
         }
        virtual void Foo() {
                cout << "A::Foo()" << endl;
         }
};

class B : public A {
public:
        B() {
                this->Foo();
         }
        virtual void Foo() {
                cout << "B::Foo()" << endl;
         }
};

int main(int, char**)
```

```
                {
                        B       objectB;

                        return 0;
                }
```

What output does this program generate as shown? Why?

```
                #include <iostream>

                using namespace std;

                 class A {
                public:
                        A() {
                                 cout << "A::A()" << endl;
                         }
                        ~A() {
                                 cout << "A::~A()" << endl; throw
"A::exception";
                         }
                };

                class B {
                public:
                        B() {
                                 cout << "B::B()" << endl; throw
"B::exception";
                         }
                        ~B() {
                                 cout << "B::~B()";
                         }
                };

                int main(int, char**) {
                        try {
                                cout << "Entering try...catch block" <<
endl;

                                A       objectA;
                                B       objectB;

                                cout << "Exiting try...catch block" <<
endl;
                        } catch (char* ex) {
                                cout << ex << endl;
                        }

                        return 0;
                }
```

Write a C program to reverse the words in a given string without instantiating another string buffer (e.g. "I love xyz very much" becomes "much very xyz love I") (HINT)
Write a C program to count the number of times string B appears in string A. Discuss some of the possible problems.
Write a function to compress one row of an image. Assume every value is repeated. Given a pointer to the row of color values and the length, without instantiating another image buffer, replace the color values with the number of color values in a series followed by the color value. Return the new length of the array (e.g., [50 50 50 251 251 50 50 75 75 75 75] length = 11; becomes [3 50 2 251 2 50 4 75] with a length = 8)
Write a function: given a char, return the bits in reversed sequence. Emphasize speed. (HINT)
Write the C++ class declaration for the following: get, set, append, insert, remove pointers in an array. Compare results, methods, and requirements between using a linked list template and an array.
Write the output of the given C++ program. The program has base class A and derived class B, constructors, virtual destructors, virtual functions, and non-virtual functions. The program instantiates a new A, new B, and an A pointer with a new B. Each function has a cout in it. (DETAILS)
What is the difference betwen a virtual function and regular function in C++?
Write the class(es) for a linked list.
Write code for traversing a binary tree.
What is a static variable in C++?
If you had a class with the method "draw," and wanted sometimes to draw a circle, sometimes a rectangle, and sometimes a line, what would you do? What if I wanted to draw two different shapes? (HINT)
What are some cross-platform issues?
What are some distributed processing issues?
What are some Multi-threaded issues?
What artifact is shown when a wavelet is applied to an image represented by one long row, instead of a series of rows and columns? (This was part of the interviewer's Master's thesis, I'm sure...)
What is multiple inheritance? Is it to be avoided? Why? How do you avoid it?
In MFC, describe the Document/View architecture. What is the first thing that happens at start-up?

Technical Interview Questions For Computer Science Programmers

The following is a list of questions which have been floating around some
Internet circles. They are excellent prep questions for a technical interview.

I haven't made any attempt to clean them up, spelling or organization wise.

Don't bother asking me for the answers,
1) I don't have an answer key.
2) It'd do you good to think.

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 subarray 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 32 bit number. (caution: looping through testing each bit is not a solution).

10. What are the different ways to say, 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=0
else
        y =x
```

There is a logical, arithmetic and a datastructure soln to the above problem.

11. Reverse a linked list.

12. Insert in a sorted list

13. In a X's and 0's game (i.e. TIC TAC TOE) if you write a program for this give a gast way to generate the moves by the computer. I mean this should be the fasteset 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.

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 optimise 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 interv. 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 algo 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 - sol given in the C lib -> typec.h)

41. Fundas 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. Tell me the courses you liked and why did you like them.

45. Give an instance in your life in which u were faced with a

problem and you tackled it successfully.

46. What is your ideal working environment. (They usually
    to hear that u can work in group also.)

47. Why do u think u are smart.

48. Questions on the projects listed on the Resume.

49. Do you want to know any thing about the company.(Try to ask some
    relevant and interesting question).

50. How long do u want to stay in USA and why?

51. What are your geographical preference?

52. What are your expecctations from the job.

53. 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.

54. Do a breadth first traversal of a tree.

55. Write code for reversing a linked list.

56. 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).

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
    deleting an element from a linked listy
    traversing a linked list
    efficient way of elimiating duplicates from an array

63. what are various problems unique to distributed databases

64. declare a void pointer
    a)   void   *ptr;

65. make the pointer aligned to a 4 byte boundary in a efficient manner
    a)   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
    els, 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.

```
                       O P
                       |
                       |
                       O B
                      / \
                     /   \
                    /     \
                   O ?     O ?
```

       determine where do A and C go

69. Describe the file system layout in the UNIX OS
    a) describe boot block, super block, inodes and data layout

70. In UNIX, are the files allocated contiguous blocks of data
    a)   no, they might be fragmented
    how is the fragmented data kept track of
    a)   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.
     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 performace 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 sqaures and
     they are shuffled.  Write a program to rearrange the 16 squares to
     get the original big square.
```

---

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.

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 0'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 << 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 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.

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 <= 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.

```
0  P
|
|
0  A
|
|
0  B
|
|
0  C
```

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

```
        0  P
        |
        |
        0  B
       /  \
      /    \
     /      \
   0  ?      0  ?
```

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, log2(2) =1, log2(3)=1, log2(4)=2..... But this macro does not work out log2(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 >= 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 simples 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 occurance 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

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.

Question: How could you determine if a linked list contains a cycle in it, and, at what node the cycle starts?

Answer: There are a number of approaches. The approach I shared is in time N (where N is the number of nodes in your linked list). Assume that the node definition contains a boolean flag, `bVisited`.

```
struct Node
{
  ...
  bool bVisited;
};
```

Then, to determine whether a node has a loop, you could first set this flag to `false` for all of the nodes:

```
// Detect cycle
// Note: pHead points to the head of the list (assume already
exists)
Node *pCurrent = pHead;
while (pCurrent)
{
  pCurrent->bVisited = false;
  pCurrent = pCurrent->pNext;
}
```

Then, to determine whether or not a cycle existed, loop through each node. After visiting a node, set `bVisited` to `true`. When you first visit a node, check to see if the node has already been visited (i.e., test `bVisited == true`). If it has, you've hit the start of the cycle!

```
bool bCycle = false;
pCurrent = pHead;
while (pCurrent && !pCycle)
{
  if (pCurrent->bVisited == true)
    // cycle!
    pCycle = true;
  else
  {
    pCurrent->bVisited = true;
    pCurrent = pCurrent->pNext;
  }
}
```

A *much better* approach was submitted by 4Guys visitor George R., a Microsoft interviewer/employee. He recommended using the following technique, which is in time O(N) and space O(1).

Use two pointers.

```
// error checking and checking for NULL at end of list omitted
p1 = p2 = head;

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

`p2` is moving through the list twice as fast as `p1`. If the list is circular, (i.e. a cycle exists) it will eventually get around to that sluggard, `p1`.

Thanks George!

Question: How would you reverse a doubly-linked list?

Answer: This problem isn't too hard. You just need to start at the head of the list, and iterate to the end. At each node, swap the values of `pNext` and `pPrev`. Finally, set `pHead` to the last node in the list.

```
Node * pCurrent = pHead, *pTemp;
while (pCurrent)
{
  pTemp = pCurrent->pNext;
  pCurrent->pNext = pCurrent->pPrev;
  pCurrent->pPrev = temp;

  pHead = pCurrent;

  pCurrent = temp;
}
```

Question: Assume you have an array that contains a number of strings (perhaps `char * a[100]`). Each string is a word from the dictionary. Your task, described in high-level terms, is to devise a way to determine and display all of the anagrams within the array (two words are anagrams if they contain the same characters; for example, `tales` and `slate` are anagrams.)

Answer: Begin by sorting each element in the array in alphabetical order. So, if one element of your array was `slate`, it would be rearranged to form `aelst` (use some mechanism to know that the particular instance of `aelst` maps to `slate`). At this point, you `slate` and `tales` would be identical: `aelst`.

Next, sort the entire array of these modified dictionary words. Now, all of the anagrams are grouped together. Finally, step through the array and display duplicate terms, mapping the sorted letters (`aelst`) back to the word (`slate` or `tales`).

Question: Given the following prototype:

```
int compact(int * p, int size);
```

write a function that will take a sorted array, possibly with duplicates, and compact the array, returning the new length of the array. That is, if `p` points to an array containing: `1, 3, 7, 7, 8, 9, 9, 9, 10`, when the function returns, the contents of `p` should be: `1, 3, 7, 8, 9, 10`, with a length of 5 returned.

Answer: A single loop will accomplish this.

```
int compact(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++;
}
```

Happy Programming!

Riddles

- Why is a manhole cover round?
- How many cars are there in the USA? (A popular variant is "How many gas stations are there in the USA?")
- How many manhole covers are there in the USA?
- 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?
- 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?

- Imagine a disk spinning like a record player turn table. Half of the disk is black and the other is white. Assume you have an unlimited number of color sensors. How many sensors would you have to place around the disk to determine the direction the disk is spinning? Where would they be placed?
- Imagine an analog clock set to 12 o'clock. Note that the hour and minute hands overlap. How many times each day do both the hour and minute hands overlap? How would you determine the exact times of the day that this occurs?
- 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?
- Pairs of primes separated by a single number are called prime pairs. Examples are 17 and 19. Prove that the number between a prime pair is always divisible by 6 (assuming both numbers in the pair are greater than 6). Now prove that there are no 'prime triples.'
- There is a room with a door (closed) and three light bulbs. Outside the room there are three switches, connected to the bulbs. You may manipulate the switches as you wish, but once you open the door you can't change them. Identify each switch with its bulb.
- Suppose you had 8 billiard balls, and one of them was slightly heavier, but the only way to tell was by putting it on a scale against another. What's the fewest number of times you'd have to use the scale to find the heavier ball?
- 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?
- You have 4 jars of pills. Each pill is a certain weight, except for contaminated pills contained in one jar, where each pill is weight + 1. How could you tell which jar had the contaminated pills in just one measurement?
- The SF Chronicle has a word game where all the letters are scrambled up and you have to figure out what the word is. Imagine that a scrambled word is 5 characters long:
    1. How many possible solutions are there?
    2. What if we know which 5 letters are being used?
    3. Develop an algorithm to solve the word.
- There are 4 women who want to cross a bridge. They all begin on the same side. You have 17 minutes to get all of them across to the other side. It is night. There is one flashlight. A maximum of two people can cross at one time. Any party who crosses, either 1 or 2 people, must have the flashlight with them. The flashlight must be walked back and forth, it cannot be thrown, etc. Each woman walks at a different speed. A pair must walk together at the rate of the slower woman's pace.

  Woman 1: 1 minute to cross

  Woman 2: 2 minutes to cross

  Woman 3: 5 minutes to cross

  Woman 4: 10 minutes to cross

  For example if Woman 1 and Woman 4 walk across first, 10 minutes have elapsed when they get to the other side of the bridge. If Woman 4 then returns with the flashlight, a total of 20 minutes have passed and you have failed the mission. What is the order required to get all women across in 17 minutes? Now, what's the other way?

- If you had an infinite supply of water and a 5 quart and 3 quart pail, how would you measure exactly 4 quarts?
- 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?
- If you have two buckets, one with red paint and the other with blue paint, and you take one cup from the blue bucket and poor it into the red bucket. Then you take one cup from the red bucket and poor it into the blue bucket. Which bucket has the highest ratio between red and blue? Prove it mathematically.

Algorithms

- What's the difference between a linked list and an array?
- Implement a linked list. Why did you pick the method you did?
- Implement an algorithm to sort a linked list. Why did you pick the method you did? Now do it in O(n) time.
- Describe advantages and disadvantages of the various stock sorting algorithms.
- Implement an algorithm to reverse a linked list. Now do it without recursion.
- Implement an algorithm to insert a node into a circular linked list without traversing it.
- Implement an algorithm to sort an array. Why did you pick the method you did?
- Implement an algorithm to do wild card string matching.
- Implement strstr() (or some other string library function).
- Reverse a string. Optimize for speed. Optimize for space.
- Reverse the words in a sentence, i.e. "My name is Chris" becomes "Chris is name My." Optimize for speed. Optimize for space.
- Find a substring. Optimize for speed. Optimize for space.
- Compare two strings using O(n) time with constant space.
- Suppose you have an array of 1001 integers. The integers are in random order, but you know each of the integers is between 1 and 1000 (inclusive). In addition, each number appears only once in the array, except for one number, which occurs twice. Assume that you can access each element of the array only once. Describe an algorithm to find the repeated number. If you used auxiliary storage in your algorithm, can you find an algorithm that does not require it?
- Count the number of set bits in a number. Now optimize for speed. Now optimize for size.
- Multiple by 8 without using multiplication or addition. Now do the same with 7.
- Add numbers in base *n* (not any of the popular ones like 10, 16, 8 or 2 -- I hear that Charles Simonyi, the inventor of Hungarian Notation, favors -2 when asking this question).
- Write routines to read and write a bounded buffer.
- Write routines to manage a heap using an existing array.
- Implement an algorithm to take an array and return one with only unique elements in it.
- Implement an algorithm that takes two strings as input, and returns the intersection of the two, with each letter represented at most once. Now speed it up. Now test it.
- Implement an algorithm to print out all files below a given root node.
- Given that you are receiving samples from an instrument at a constant rate, and you have constant storage space, how would you design a storage algorithm that would allow me to get a representative readout of data, no matter when I looked at it? In other words, representative of the behavior of the system to date.
- How would you find a cycle in a linked list?

- Give me an algorithm to shuffle a deck of cards, given that the cards are stored in an array of ints.
- The following asm block performs a common math function, what is it?

-
  ```
  cwd xor ax, dx
  sub ax, dx
  ```
- Imagine this scenario:
  I/O completion ports are communictaions ports which take handles to files, sockets, or any other I/O. When a Read or Write is submitted to them, they cache the data (if necessary), and attempt to take the request to completion. Upon error or completion, they call a user-supplied function to let the users application know that that particular request has completed. They work asynchronously, and can process an unlimited number of simultaneous requests.
  Design the implementation and thread models for I/O completion ports. Remember to take into account multi-processor machines.
- Write a function that takes in a string parameter and checks to see whether or not it is an integer, and if it is then return the integer value.
- Write a function to print all of the permutations of a string.
- Implement malloc.
- Write a function to print the Fibonacci numbers.
- Write a function to copy two strings, A and B. The last few bytes of string A overlap the first few bytes of string B.
- How would you write qsort?
- How would you print out the data in a binary tree, level by level, starting at the top?

Applications

- How can computer technology be integrated in an elevator system for a hundred story office building? How do you optimize for availability? How would variation of traffic over a typical work week or floor or time of day affect this?
- How would you implement copy-protection on a control which can be embedded in a document and duplicated readily via the Internet?
- Define a user interface for indenting selected text in a Word document. Consider selections ranging from a single sentence up through selections of several pages. Consider selections not currently visible or only partially visible. What are the states of the new UI controls? How will the user know what the controls are for and when to use them?
- How would you redesign an ATM?
- Suppose we wanted to run a microwave oven from the computer. What kind of software would you write to do this?
- What is the difference between an Ethernet Address and an IP address?
- How would you design a coffee-machine for an automobile.
- If you could add any feature to Microsoft Word, what would it be?
- How would you go about building a keyboard for 1-handed users?
- How would you build an alarm clock for deaf people?

Thinkers

- How are M&Ms made?

- If you had a clock with lots of moving mechanical parts, you took it apart piece by piece without keeping track of the method of how it was disassembled, then you put it back together and discovered that 3 important parts were not included; how would you go about reassembling the clock?
- If you had to learn a new computer language, how would you go about doing it?
- You have been assigned to design Bill Gates bathroom. Naturally, cost is not a consideration. You may not speak to Bill.
- What was the hardest question asked of you so far today?
- If MS told you we were willing to invest $5 million in a start up of your choice, what business would you start? Why?
- If you could gather all of the computer manufacturers in the world together into one room and then tell them one thing that they would be compelled to do, what would it be?
- Explain a scenario for testing a salt shaker.
- If you are going to receive an award in 5 years, what is it for and who is the audience?
- How would you explain how to use Microsoft Excel to your grandma?
- Why is it that when you turn on the hot water in any hotel, for example, the hot water comes pouring out almost instantaneously?
- Why do you want to work at Microsoft?
- Suppose you go home, enter your house/apartment, hit the light switch, and nothing happens - no light floods the room. What exactly, in order, are the steps you would take in determining what the problem was?
- Interviewer hands you a black pen and says nothing but "This pen is red."