**2nd LEVEL INTERVIEW PROCESS**  
1. Software test: One hour Software test. U will be placed on a system with an application opened that is containing bugs in it. A reference doc will be provided where u will find the description of six modules of the application. U have to find maximum number of bugs in those six modules given in the reference doc in one hour. Each module contains at least one bug. U will have to keep writing the bug in a paper provided to u as u keep finding it with the time when u found it.  
  
2. 5 rounds of Interviews each of approximately 1 hour.  
  
1st will be by a Senior Team member level guy, focusing on everything from simple codes to find errors in them, writing simple algos, giving very simple puzzles, test cases for a marker, ur projects, about urself, etc. The guy will keep giving hints and help u to solve the problem.  
  
2nd will be by a Manager level guy, based on ur technical skills, puzzles to be solved on the whiteboard in front of him, algo to reverse a string using array, questions on ur projects, test cases of a duster, the projects u have mentioned in ur CV. More of a question-answer based approach, not very interactive.   
  
3rd will be by a senior HR on why Adobe, what keeps u going, where r u placed in ur organization, all HR questions and about ur projects and Organizational levels.  
  
4th will be by a Senior Team member level guy, focusing only on puzzles, lots of puzzles and scenario based test cases, like how to test an imaging application that removes the red eye affect from an image, test cases of an VOIP phone. The guy will keep giving hints and help u to solve the problem.  
  
5th will be by a Senior Team member level guy, focusing on problem solving approach, scenario based error investigation, like an attachment is not opening in outlook on a particular system, what all can be the reasons. The guy will sort of discuss with u the problem and will try to find out the solution at the same time assessing ur problem solving skill. Then he will play a logical game, then some general questions, test cases for a radio, something about Adobe etc.

**Interview2 :**  
He was friendly at the start but this interview was my worst. He asked me my favorite subject. I said that it was Programming. (He laughed at that)  
  
1. Which are the four storage classes in C.  
     Answer : static, extern, register, auto  
        
2. Given a program:  
  
      int i;  
     int main()  
     {  
        int j;  
        int \*k = (int \*) malloc (sizeof(int));  
        ...  
     }  
     Where are each of these variables stored?  
  
     Answer : I started off correctly, but he was able to confuse me. He brought in shared libraries, static libraries fundas into the discussion. We had a discussion for about twenty-minutes on this. Finally, he was happy with one of my answers because I had deduced which policy made sense and answered correctly. He said that out of all the people interviewed so far (I was second last), nobody had been able to answer all of these questions correctly  
       
3. Question on polymorphisms.  
     this is easy - get it from any C++ book. He tried to confuse me again, but this time I was ready and he was finally satisfied.  
  
     Then he looked at my grades and said that out of all your grades, you have only two Bs and one of them is in Compilers. Why? (Damn it.. three non-A grades and that's all they ask about. What's wrong with this world?!)  
  
     Didn't you like Compilers? "Not in particular", I replied. "Fine. Now, I HAVE to ask you questions on compilers", he said.  
       
4. He again went back to the first question he had asked me. Once again, I had no satisfactory answer for him.  
        
5. Then he wrote out some code and asked me how the compiler will generate code for it. I gave some answer, but he was clearly not satisfied. I thought it was all over by then.  
  
     Then, he asked me a DIP question. He commented that he had given that particular question to his juniors in IITB once and they had done a very good job at it. The problem is this:  
  
     A teacher is writing on a blackboard and we want to pass all the information on the blackboard over a low-bandwidth network in real-time. How do we do it.  
  
     Answer : I first suggested that we capture only a small portion of the board. To locate that portion, we could search for the chalk in the prof's hand - of course, taking care that it had the blackboard in the background (no point capturing a video of the prof scratching his chin, na?). Further, if the prof was writing only text, we could convert the video into text by OCR and then transmitting. Simple diagrams could also be reduced to a set of vector-graphics instructions (we rarely, see the prof shading stuff). I think he liked my approach, but was not completely satisfied. Anyway, we left it at that and went forward.  
        
6. Given a set of words one after another, give me a data structure so that you'll know whether a word has appeared already or not.  
  
     Answer : I suggested various alternatives. but he kept helping me and finally, we came up with an array of pointers to 26-trees (each node of the tree has 26 children). Store every word as a path from the root to a leaf with pointers in the correct places. For example, hello would be stored as - pointer from 'h' index of the root array to a node which had a pointer from 'e' index of it's array to a node which had a pointer from 'l' index of the array.. and so on. This is both time and space efficient.  
  
7. He asked me some questions on Interprocess Communication: What's a semaphore? How are they used? He would often pick out words from my answers and ask me what they meant. He wanted to make sure that I really knew what I was talking about. I was able to answer all his questions, but I made the mistake of telling him, when we started off that I didn't know much about this subject as I had done it a long time ago. He was very annoyed at that, apparently because a lot of people before me had said this.  
  
8. He then asked me some DB fundas. Transaction. Serializability, Consistent state, etc. I was able to answer all of them. I stumbled around a bit in a few questions where I was explaining correctly, but not using the keywords that he was looking for.  
        
9. Finally, he asked me whether I had any questions. I thought that I should say something to make him realize that I was not completely stupid and so asked him whether there was any logic to the order in which the short-listed candidates were called. This turned out to be a dumb move. The order was alphabetic and he sent me off with a parting shot, saying "You guys do pattern recognition and stuff and still you can't recognize such a simple pattern" Me and my big mouth! Moral of the story: Don't ask questions for the sake of asking.  
  
Interview3 : Puzzle round  
After the first two interviews, this one was like having a warm batch after being cold and wet for days! I did well in this one.  
  
1. There is a clock at the bottom of the hill and a clock at the top of the hill. The clock at the bottom of the hill works fine but the clock at the top doesn't. How will you synchronize the two clocks. Obviously, you can't carry either of the clocks up or down the hill! And you have a horse to help you transport yourself. And, the time required for going up the hill is not equal to the time required to go down the hill.  
  
     Answer : You have to go up the hill and come back, with horse, without horse, getting four equations to solve four unknowns - time to go uphill - with horse, without horse, time to go downhill - with horse, without horse. Then you can go up the hill and set the clock to '(time when you left) + (time to go uphill with horse)'  
        
2. There was one more puzzle.. I don't remember it. but I do remember that we started discussing ways of generating large prime numbers. I told him the funda of Mersenee primes (luckily remembered it) and he was decently impressed.  
  
3. We also talked a bit about my phone browser project.  
  
Finally, two people were selected out the ten that were shortlisted.