PAPER: ADOBE SYSTEM PLACEMENT PAPER FOR FRESHERS (WRITTEN & INTERVIEW )

Written rounds:

Round 1:  Aptitude - Trivially simple. Time could be a problem for candidates not preparing for CAT. Fifteen questions in fifteen minutes.

Round 2: Quantitative - This was even more trivial. Thirty questions in thirty minutes.

These two rounds were entirely objective, multiple-choice with no negative marking. We had to fill in an OMR sheet and very few of us had pencils and erasers with us - almost everybody had pens. That was bad as we couldn't change an option after marking it.

Round 3: Engineering. This included a lot of general stuff. Don't remember much about it. Thirty minutes. 15 marks

Round 4:  Code. This was a real test. A lot of questions and quite a few that I hadn't seen before. We were supposed to write working C code in a limited time. 15 questions. 45 marks. It was like Neo fighting Smith in Matrix 3!

10 were selected out of 60 people who wrote the test.

Interview1:
He is very soft-spoken and one of my friend told me that this guy had brutally destroyed him. So I was terribly nervous. I usually am confident but in that room, I was stammering! Finally, it turned out that he wasn't that brutal after all. Moral of the story: Stay confident at all times.

  1. He asked me my specialization? When I told him that I had none as I didn't want to specialize in this stage, he was a little surprised but appeared satisfied with my reason.

   2. Why not further studies? (He had noted that I was third in my batch. He appeared impressed by that

      Answer : I told him that my profile clearly indicated that I've been trying to get into the industry via internships, industry-funded projects right from second year, second sem. I said that I was fully sure that I didn't want to do MS anytime soon.

3. He asked me to tell him about my favorite project.

      Answer : I told him about the web-browser that I had developed for cell-phones. I thought that was the only project which was closest to what Adobe was working on. He appeared satisfied with my answers.

   4. He then looked at my grades. He commented that my lowest grade - B- was in Digital Image Processing. I just looked at him like a doofus thinking of what to say. But he quickly added, 'don't worry, it happens.' So people, be fully prepared to explain any anomalous grades. I was prepared with the explanation of the W in the my grade-sheet but not of the B- in DIP. I know that this is really stupid considering that I was interviewing with Adobe. Don't make this mistake.

   5. He then asked me a question that had been asked in Round 4, written test:

      Describe an optimal algorithm to find the second minimum number in an array of numbers. What is the exact number of comparisons required in the worst case? Note that they didn't ask the order in Big-Oh notation. They wanted the exact number of comparisons.

      Answer : I screwed up, big time in this question. I had superficially discussed this question with my friend a while ago and he had outlined an algorithm which I thought that I had understood, but I hadn't. I started off explaining it but got stuck in the middle. He sternly told me to read it up again. One solution that I could tell him, and which I had written in the test was this:

      Use two variables  - min and second min. Initialize them by comparing the first two elements of the array. This is (1) comparison. Then, go through the entire array, from index 2 to n-1 comparing each element, first with min and then with second min, updating each variable as necessary. This will involve a worst case of two comparisons for each element. Therefore, total number of comparisons = 2\*(n-2) + 1 = 2\*n - 3 comparisons.

      I'll try to update this with a better solution, sometime soon.

   6. Given a polygon (could be regular, irregular, convex, concave), find out whether a particular point lies inside it or outside it.

      Answer : This is an easy, straight question from graphics. You shoot a ray parallel to the x-axis passing through this point. Start with odd parity. Change parity of ray each time it intersects an edge of the polygon (consider special case of when the line passes through a vertex of the polygon. Change parity only if it passes through a vertex which has one edge above it and one edge below the ray). If the parity of ray is even when it passes through the point, it is inside the polygon, else it is not.

   7. He asked me to explain Canny's algorithm to him. (this was because my DIP project was related to this)

      Answer :  This is simple. Study DIP

   8.  Then, he gave me a practical problem to solve: Suppose you are given an image which contains some text and some photos. How do you find the location of the image?

      Answer : I gave various alternatives - from searching for RGB components, to using OCR.. he didn't appear fully satisfied. I think he was looking for edge-detection, but that would fail, if the text contained tables, 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 Inter process 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 fundamental of Mersenee primes (luckily remembered it) and he was decently impressed.
   3. We also talked a bit about my phone browser project.