MICROSOFT

Rajiv Sharma

Process Structure:

1. Aptitude round
2. Programming round
3. Wildcard GD or case study round
4. Technical interviews
5. HR interview

1. Aptitude Round:

This is the first round of the process and everyone who has successfully applied for the company are eligible for this round. There were roughly around 30 MCQs. Like most technical aptitudes it focuses on the following domains:

- C programming language (20-25 MCQs):

Some companies don't make any strict regulations regarding the language in question, but Microsoft always goes with C/C++. The questions are similar to what you would find in books like "Let us C". Questions regarding pointer manipulation, predicting the output, pre-order post-order increment decrement, structures, arrays formed the majority.

- Data Structures and Algorithms, OS and DBMS (5-10 MCQs):

These are their favourite areas of computer science that they choose to focus on. That being said the questions asked in this section were quite few. Maybe 5 or less than that. Questions regarding like memory allocation in OS, preferred data structure for a particular application, properties of database management systems made it in.

TIP:

1. While preparing for the aptitude, sharpen your C skills. Give second preference to OS, DBMS and DSA.
2. Get extensive practice in the most common types of C aptitude questions and give regular online tests.
3. Cover what you can in OS, DBMS and DSA in the time that remains.

2. Programming Round:

Once you clear the aptitude you become eligible for the programming round. This round has two questions meant to be solved in C. The focus is on the following aspects and in this particular order

- Correctness
  - o Output is correct for the given test cases
- Completeness
  - o Output is correct for all the test cases
- Time Complexity
  - o The program has the specified time complexity or less
- Space Complexity
  - o The program does not occupy too much excess space

You don’t have to write the entire programs just the necessary functions to perform the tasks given. The results were declared after a few days when Microsoft visited:

Q1. Find an element in a rotated sorted array in O(log n) time.
    For ex. Find 4 in 7 8 9 1 2 4 5

Q.2 With a parent pointer in every node of a binary tree, find the inorder successor of a given node.

I solved both of these questions, yet my solution for the first one used some amortized amount of time (amortized time is time that is not significant enough to affect the order, that is log n, but it is not too insignificant to be ignored as constant) more than the required time constraint which had me worried a little.
TIP: Work on getting at least the right answer, worry about the run time and space complexity later. Even people who could get just the right answer made it to the wildcard round

3. Wildcard GD or case study round:
People who barely managed to get their output right in the last round were eligible for this round. I don’t have details about how this round was conducted since I wasn’t part of it but it is pretty standard procedure, the same kind of GD or case study that happens in most companies.

4. Technical interviews:
Everyone who made it past the programming round in a single attempt or through the wildcard round were called in for the tech interviews. I had a single tech interview but some people were called in as many as 3 times. The questions asked were as follows:
(The interviewer took my resume in his hand)
1. Interviewer: While I read the resume go ahead and introduce yourself.
Me: Keep it real. Say stuff that is good about you AND makes you a good candidate for the job. You never have more than 3-4 minutes for such questions so focus on conveying what is most important. After general introductions, I talked about how passionate I am about programming and technology in general, how I didn’t just see it as a subject or a career option but something that I enjoyed doing and have successfully used in my daily life. Talk about what values and work ethics you believe in. End it with what qualities you have as a person or as a possible employee. Please note that you should not leave the interviewer on just your words to believe on. Give short descriptive examples to support what you are saying but don’t go on telling long boring stories.
Example: Sir I have always made a sincere effort to use my coding capabilities in my day to day life and not just in the classroom. I made an application for syncing subtitles which I use whenever I find that my videos and subtitles are out of sync.

2. Interviewer: Questions about DSA and DBMS
Me: I gave the required answers. I am not elaborating on this part since this is something that you can answer just by knowing your stuff. It was nothing out of the ordinary but it wasn’t all that easy. DSA and DBMS were among my favourite subjects so I did pretty well.

3. Interviewer: Write code for the subtitle syncing application you talked about, not the entire thing, just the crux of it.
Me: I wrote the code since I had actually made the application. This is an important question where he was testing whether I told the truth or not earlier. So whatever happens do not lie or make up even a single detail while explaining something, you never know what might come back to haunt you.

4. Interviewer: Write code for designing the ADT (Abstract Data Type) for all the classes that might be required to represent the game of chess
Me: This was the kind of question that I wasn't expecting. Also the reason I got this question was that I had mentioned chess in my hobbies. So keep in mind every small thing you put in your resume matters a lot. He allowed me to choose my language of choice so I began with Java and then I did the important part.
I articulated my thoughts. Something like "Hmm…It would be foolish to make a class for each and every chess piece and the players and repeat multiple properties across the classes. It is not only space consuming but also a poor design. Maybe I should start with the interfaces…for that I need to figure out what are the common elements in all the pieces? Ok every piece has a colour black or white, they have set of valid and invalid moves, they are either alive, that is, on the board or they are dead, they would have a location describing where the piece is currently at" …and so on. I went step by step through the process and it was much slower than it would have been if I directly started writing the code. But it was worth it for 2 reasons:
o If you make a mistake midway the interviewer can and will help you with it because you have been talking him through your entire code and now he has complete idea of what is going on
o If you do something out of the ordinary, like you decide to implement a particular design pattern for the problem after talking about the reasons why you choose it in particular, your continuous talking will make sure that this shining moment of yours doesn't go unnoticed. This makes a lot of difference in the end.
This question went on for about an hour (not kidding), and finally the interviewer let me go. After this I was directly called for the HR interview.

5. HR interview:

Interviewer: Why Microsoft?
Me: Give some answer that is well thought and prepared beforehand.

Interviewer: You left Barclays for Microsoft?
Me: Basically what they wanted to know here is that I don't have the same plans for Microsoft. I told them Barclays was an investment bank and while I was fortunate to get a chance to work in that great environment I want to start my career in a product based company so Microsoft (Microsoft is becoming a service based company now BTW)
Note: If you have done internship somewhere else and have rejected an offer from there, say only good things about your previous employer. Anything else will reflect badly on you.

Interviewer: What makes you better than you colleagues?

Me: Whatever nice things you can say about yourself, everything comes here. Don’t, however mention the negative points of your colleagues, in hopes of making yourself look good. It will only make you look bad.

Interviewer: Write the artificial intelligence logic in code for your Chess representation
Me: Now the fact that this question was asked in HR rather than Tech should give you an idea about the real purpose of this question. When this question was asked we didn’t even have AI as a subject yet. The interviewer knew beforehand that I could not code this. He wanted to see how I would handle it. So after thinking for a few minutes I decided to draw flow charts and map out scenarios. Here is what I mean:
Case 1: Player 1 turn with the possibility of killing off a chess piece belonging to the opponent:

When not to kill:
a. When the piece being killed by player 1 has lesser value than his own piece. Example: My Queen killing the opponent’s pawn doesn’t make sense

b. When the piece used to kill off the opponents piece is in some way protecting the King from a probable check
And so on including cases for when to kill the opponent’s piece. In essence I laid out the rules that an AI agent could use to play well if I could code the AI agent. The interviewer liked this approach

That was it. This was the last round and then I was done with my interviews. Thank you for reading and best of luck :-)