The *IBM Revised Programmer Aptitude Test*, developed by J.L. Hughes and W.J. McNamara, is a relatively straightforward standard-type test of IQ and mathematical ability. It consists of three parts.

Part I: Number Series

This section involves identifying the 7th term in a number sequence. There are **26** questions to be completed in just **10** minutes. The correct answers are:

Although the questions are not particularly difficult, many require the test-taker to evaluate more than one hypothesis, which takes time. The time constraint effectively filters out candidates who are not quick in their thinking. While the connection to actual programming skills is debatable, IBM could afford to use such a rigid selection method given the volume of applicants. The questions get progressively harder, but once you figure out their particular style and pattern, they are not difficult. They are surprisingly similar in design, and when you realise that each number depends only on a single predecessor (no Fibonacci-type series etc., unlike many similar tests), solving gets faster.

Part II: Figure Analogies

This section consists of visual analogies, where one must identify the correct transformation of geometric figures. These are classical IQ-test figure analogies, and after a few you get the pattern behind them. There are **40** questions with a **20**-minute time limit. The correct answers are:

2 4 5 4 4 4 4 1 2 4 5 3 2 3 3 4 4 5 1 2 3 3 1 3 1 5 1 3 1 3 1 4 5 3 4 5 3 2 4 5

Given the number of questions and the cognitive demand of testing multiple visual hypotheses for the more advanced questions, the time is definitely insufficient. Again, the test prioritises speed over depth, aiming to eliminate slower thinkers rather than measure deep reasoning ability.

Part III: Simple Arithmetic

In the final section, **20** questions are to be answered in **30** minutes, which proves more than sufficient. The questions are basic arithmetic problems, solvable with lower secondary school mathematics (grades 7-9) knowledge, with the exception of question 14. The correct answers are:

Most questions involve office-related scenarios framed in mainframe computing language, such as calculating the output of a card reader processing a set number of cards per hour. The section poses little challenge and requires no abstract reasoning. Thus, it does not measure programming aptitude.

Overall Assessment

The test suffers from a significant imbalance in time allocation across its sections. Despite being labelled a *programming* aptitude test, it contains no questions that directly test programming skills or computer knowledge. Nevertheless, it offers a glimpse into historical recruitment practices in computing, prioritising mental calculating and processing speed over domain-related skills.

The author of this note completed the parts in 17, 26, and 14 minutes respectively, clearly overshooting the time limit for two of the three parts, and was thus probably too slow to be accepted as an IBM trainee. In published research, four groups of a total of over 270 prospective programmers had average (standard deviation) scores of 60 (9), 58 (9), 53 (13), and 51 (13), thus with the best 15% at 69+, 67+, 66+, and 64+ respectively, out of a max of 86. While not marking the time limits exactly, the author missed at least 8+7+0 = 15 questions, all due to the limits.

References

Hughes, J. L., & McNamara, W. J. (1959). *The revised programmer aptitude test*. IBM Corporation, New York, NY.

Hughes, J. L., & McNamara, W. J. (1961). A comparative study of programed and conventional instruction in industry. *Journal of Applied Psychology*, 45(4), 225–231.

McNamara, W. J., & Hughes, J. L. (1961). A review of research on the selection of computer programmers. *Personnel Psychology*, *14*, 39–51.