

Prathamesh Beldar

Test ID: 432006109564327 | +918459919259 | prathameshbeldar@gmail.com

Test Date: September 8, 2024

Computer Programming

75 /100



English Comprehension

63 /100



Quantitative Ability

76 /100



Logical Ability

56 /100



Automata

89 /100



Personality

✓ Completed

Computer Programming

75 / 100

Basic Programming

74 / 100

Data Structures

81 / 100

OOP and Complexity Theory

70 / 100

English Comprehension

63 / 100

CEFR: B2

Vocabulary

71 / 100

Grammar

60 / 100

Comprehension

57 / 100

Quantitative Ability

76 / 100

Number system

71 / 100

Word problems

81 / 100

Logical Ability

56 / 100

Inductive Reasoning

Deductive Reasoning

57 / 100

56 / 100

Automata

89 / 100

Programming Ability

Programming Practices

Functional Correctness

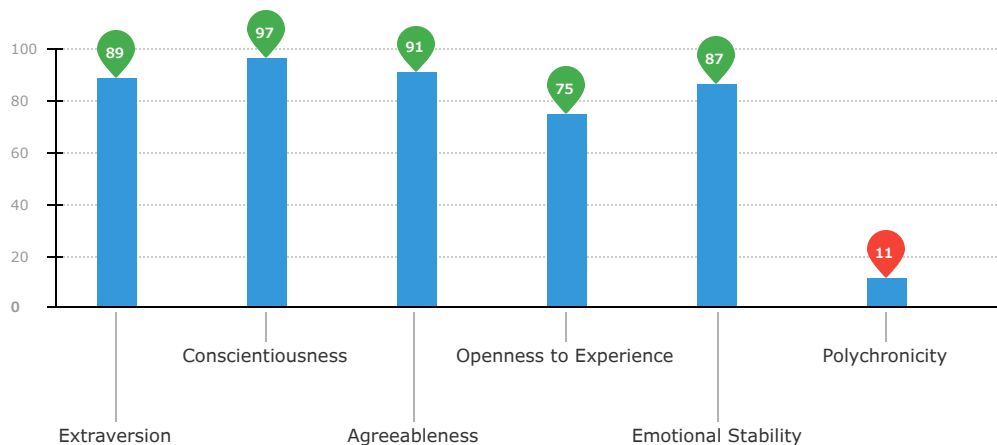
90 / 100

75 / 100

69 / 100

Personality

Completed



Competencies

People Interaction

Self-Drive

Trainability

Repetitive Job Suitability

Work attributes

1 | Introduction

About the Report

This report provides a detailed analysis of the candidate's performance on different assessments. The tests for this job role were decided based on job analysis, O*Net taxonomy mapping and/or criterion validity studies. The candidate's responses to these tests help construct a profile that reflects her/his likely performance level and achievement potential in the job role

This report has the following sections:

The **Summary** section provides an overall snapshot of the candidate's performance. It includes a graphical representation of the test scores and the subsection scores.

The **Insights** section provides detailed feedback on the candidate's performance in each of the tests. The descriptive feedback includes the competency definitions, the topics covered in the test, and a note on the level of the candidate's performance.

The **Response** section captures the response provided by the candidate. This section includes only those tests that require a subjective input from the candidate and are scored based on artificial intelligence and machine learning.

The **Learning Resources** section provides online and offline resources to improve the candidate's knowledge, abilities, and skills in the different areas on which s/he was evaluated.

Score Interpretation

All the test scores are on a scale of 0-100. All the tests except personality and behavioural evaluation provide absolute scores. The personality and behavioural tests provide a norm-referenced score and hence, are percentile scores. Throughout the report, the colour codes used are as follows:

- Scores between 67 and 100
- Scores between 33 and 67
- Scores between 0 and 33

2 | Insights

English Comprehension

63 / 100

CEFR: **B2**

This test aims to measure your vocabulary, grammar and reading comprehension skills.

You have a good understanding of commonly used grammatical constructs. You are able to read and understand articles, reports and letters/emails related to your day-to-day work. The ability to read, understand and interpret business-related documents is essential in most jobs, especially the ones that involve research, technical reading and content writing.

Logical Ability

56 / 100



Inductive Reasoning

57 / 100

This competency aims to measure your ability to synthesize information and derive conclusions.

You are able to work out rules based on specific information and solve general work problems using these rules. This skill is required in data-driven research jobs where one needs to formulate new rules based on variable trends.



Deductive Reasoning

56 / 100

This competency aims to measure your ability to learn by example, imitation or hit-and-trial and provides an indication of how creative the individual is.

It is good that you are able to draw valid inferences based on simple rules and instructions. The problems that one faces at work are generally complex and involve several factors. Hence it is important to take into account all the relevant inputs and then make the final decision.

Quantitative Ability

76 / 100

This test aims to measure your ability to solve problems on basic arithmetic operations, probability, permutations and combinations, and other advanced concepts.

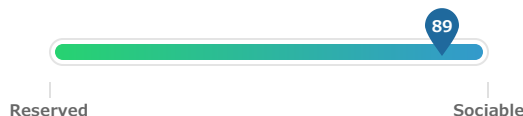
It is commendable that you are able to understand and solve complex arithmetic problems. You are able to solve basic problems of probability, logarithms, permutations, and combinations. This skill will help you in jobs where one needs to work with statistical data and make probabilistic predictions.

Personality

Competencies



Extraversion



Extraversion refers to a person's inclination to prefer social interaction over spending time alone. Individuals with high levels of extraversion are perceived to be outgoing, warm and socially confident.

- You are outgoing and seek out opportunities to meet new people.
- You tend to enjoy social gatherings and feels comfortable amongst strangers and friends equally.
- You display high energy levels and like to indulge in thrilling and exciting activities.
- You may tend to be assertive about your opinions and prefer action over contemplation.
- You take initiative and are more inclined to take charge than to wait for others to lead the way.
- Your personality is well suited for jobs demanding frequent interaction with people.



Conscientiousness



Conscientiousness is the tendency to be organized, hard working and responsible in one's approach to your work. Individuals with high levels of this personality trait are more likely to be ambitious and tend to be goal-oriented and focused.

- You value order and self discipline and tends to pursue ambitious endeavours.
- You believe in the importance of structure and is very well-organized.
- You carefully review facts before arriving at conclusions or making decisions based on them.
- You strictly adhere to rules and carefully consider the situation before making decisions.
- You tend to have a high level of self confidence and do not doubt your abilities.
- You generally set and work toward goals, try to exceed expectations and are likely to excel in most jobs, especially those which require careful or meticulous approach.



Agreeableness



Agreeableness refers to an individual's tendency to be cooperative with others and it defines your approach to interpersonal relationships. People with high levels of this personality trait tend to be more considerate of people around them and are more likely to work effectively in a team.

- You are considerate and sensitive to the needs of others.
- You tend to put the needs of others ahead of your own.
- You are likely to trust others easily without doubting their intentions.
- You are compassionate and may be strongly affected by the plight of both friends and strangers.
- You are humble and modest and prefer not to talk about personal accomplishments.
- Your personality is more suitable for jobs demanding cooperation among employees.



Openness to Experience



Openness to experience refers to a person's inclination to explore beyond conventional boundaries in different aspects of life. Individuals with high levels of this personality trait tend to be more curious, creative and innovative in nature.

- You tend to be curious in nature and is generally open to trying new things outside your comfort zone.
- You may have a different approach to solving conventional problems and tend to experiment with those solutions.
- You are creative and tends to appreciate different forms of art.
- You are likely to be in touch with your emotions and is quite expressive.
- Your personality is more suited for jobs requiring creativity and an innovative approach to problem solving.



Emotional Stability



Emotional stability refers to the ability to withstand stress, handle adversity, and remain calm and composed when working through challenging situations. People with high levels of this personality trait tend to be more in control of their emotions and are likely to perform consistently despite difficult or unfavourable conditions.

- You are calm and composed in nature.
- You tend to maintain composure during high pressure situations.
- You are very confident and comfortable being yourself.
- You find it easy to resist temptations and practice moderation.
- You are likely to remain emotionally stable in jobs with high stress levels.



Polychronicity



Polychronicity refers to a person's inclination to multitask. It is the extent to which the person prefers to engage in more than one task at a time and believes that such an approach is highly productive. While this trait describes the personality disposition of a person to multitask, it does not gauge their ability to do so successfully.

- You prefer to work on one task at a time, complete it and then move on to the next.
- You prefer orderliness and likes to concentrate on the task at hand without any distractions.
- You can find it difficult to be placed in a work environment where there is a need to multitask or where expected to engage in multiple projects simultaneously.

3 | Response

Automata



89 / 100

[Code Replay](#)

Question 1 (Language: C)

In a science research lab, combining two nuclear chemicals produces a maximum energy that is the product of the energy of the two chemicals. The energy values of the chemicals can be negative or positive. The scientist wants to calculate the sum of the energies of the two chemicals which produces maximum energy on reaction.

Write an algorithm to find the sum of the energy of the two chemicals which produces maximum energy on reaction.

Scores

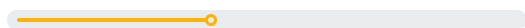
Programming Ability



80 / 100

Correct with inadvertent errors. Correct control structures and critical data dependencies incorporated. Some inadvertent errors make the code fail test cases.

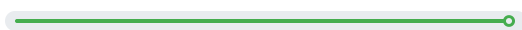
Functional Correctness



38 / 100

Partially correct basic functionality. The source code compiles and passes only some of the basic test cases. Some advanced or edge cases may randomly pass.

Programming Practices



100 / 100

High readability, high on program structure. The source code is readable and does not consist of any significant redundant/improper coding constructs.

Final Code Submitted

Compilation Status: Pass

```
1 #include <stdio.h>
2 #include <limits.h>
3
4 int main() {
5     int numOfChem;
6
7     // Read the number of chemicals
8     scanf("%d", &numOfChem);
9
10    // Ensure there are at least two chemicals
11    if (numOfChem < 2) {
12        printf("Error: There must be at least two chemicals.\n");
13        return 1;
14    }
15
16    // Variables to track the top two maximum and two minimum val
```

Code Analysis

Average-case Time Complexity

Candidate code: Complexity is reported only when the code is correct and it passes all the basic and advanced test cases.

Best case code: $O(\log N)$

*N represents number of chemicals.

Errors/Warnings

There are no errors in the candidate's code.

Structural Vulnerabilities and Errors

There are no errors in the candidate's code.

```

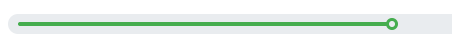
ues
17  int max1 = INT_MIN, max2 = INT_MIN;
18  int min1 = INT_MAX, min2 = INT_MAX;
19
20  // Read the energies
21  for (int i = 0; i < numOfChem; i++) {
22      int energy;
23      scanf("%d", &energy);
24
25      // Update maximum values
26      if (energy > max1) {
27          max2 = max1;
28          max1 = energy;
29      } else if (energy > max2) {
30          max2 = energy;
31      }
32
33      // Update minimum values
34      if (energy < min1) {
35          min2 = min1;
36          min1 = energy;
37      } else if (energy < min2) {
38          min2 = energy;
39      }
40  }
41
42  // Calculate the maximum product
43  int maxProduct = max1 * max2;
44  int minProduct = min1 * min2;
45
46  // Determine the maximum product and the corresponding sum
47  if (maxProduct > minProduct) {
48      printf("%d\n", max1 + max2);
49  } else {
50      printf("%d\n", min1 + min2);
51  }
52
53  return 0;
54 }

```

Test Case Execution

Passed TC: **84.62%**

Total score

 **11/13**

88%
Basic(7/8)

75%
Advance(3/4)

100%
Edge(1/1)

Compilation Statistics

2

Total attempts

2

Successful

0

Compilation errors

0

Sample failed

0

Timed out

0

Runtime errors

Response time:

00:11:15

Average time taken between two compile attempts:

00:05:38

Average test case pass percentage per compile:

42.31%

Average-case Time Complexity

Average Case Time Complexity is the order of performance of the algorithm given a random set of inputs. This complexity is measured here using the Big-O asymptotic notation. This is the complexity detected by empirically fitting a curve to the run-time for different input sizes to the given code. It has been benchmarked across problems.

Test Case Execution

There are three types of test-cases for every coding problem:

Basic: The basic test-cases demonstrate the primary logic of the problem. They include the most common and obvious cases that an average candidate would consider while coding. They do not include those cases that need extra checks to be placed in the logic.

Advanced: The advanced test-cases contain pathological input conditions that would attempt to break the codes which have incorrect/semi-correct implementations of the correct logic or incorrect/semi-correct formulation of the logic.

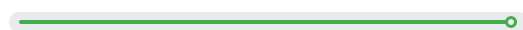
Edge: The edge test-cases specifically confirm whether the code runs successfully even under extreme conditions of the domain of inputs and that all possible cases are covered by the code

Question 2 (Language: C)

The vowels of the English alphabet are (a, e, i, o, u, A, E, I, O, U). Write an algorithm to eliminate all vowels from a given string.

Scores

Programming Ability

 **100** / 100


Completely correct. A correct implementation of the problem using the right control-structures and data dependencies.

Functional Correctness

 **100** / 100

Functionally correct source code. Passes all the test cases in the test suite for a given problem.

Programming Practices

 **50** / 100

High readability, low on program structure. The source code contains redundant/improper coding constructs and a few readability and formatting issues.

Final Code Submitted

Compilation Status: Pass

```

1 #include <stdio.h>
2 #include <ctype.h>
3
4 // Function to check if a character is a vowel
5 int isVowel(char ch) {
6     // Check for both uppercase and lowercase vowels
7     return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
8         ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U';
9 }
10
11 int main() {
12     char input[101]; // Input string, maximum length is 100 + 1 for n
13     char output[101]; // Output string
14     int j = 0;
15
16     // Read the input string
17     fgets(input, sizeof(input), stdin);
18
19     // Process each character of the input string
20     for (int i = 0; input[i] != '\0'; i++) {
21         if (!isVowel(input[i])) {
22             output[j++] = input[i];
23         }
24     }
25
26     // Null-terminate the output string
27     output[j] = '\0';
28
29     // Print the result
30     printf("%s\n", output);
31

```

Code Analysis

Average-case Time Complexity

Candidate code: Complexity is reported only when the code is correct and it passes all the basic and advanced test cases.

Best case code: $O(N)$

*N represents number of characters in the string

Errors/Warnings

There are no errors in the candidate's code.

Structural Vulnerabilities and Errors

Readability & Language Best Practices

Line 14: Variables are given very short name.

Performance & Correctness

Line 12,13: Use of hardcoded values must be avoided.

```
32 return 0;
33 }
```

Test Case Execution

Passed TC: 100%

Total score

20/20

100%

Basic(6/6)

100%

Advance(12/12)

100%

Edge(2/2)

Compilation Statistics

1

Total attempts

1

Successful

0

Compilation errors

0

Sample failed

0

Timed out

0

Runtime errors

Response time:

00:03:05

Average time taken between two compile attempts:

00:03:05

Average test case pass percentage per compile:

100%

Average-case Time Complexity

Average Case Time Complexity is the order of performance of the algorithm given a random set of inputs. This complexity is measured here using the Big-O asymptotic notation. This is the complexity detected by empirically fitting a curve to the run-time for different input sizes to the given code. It has been benchmarked across problems.

Test Case Execution

There are three types of test-cases for every coding problem:

Basic: The basic test-cases demonstrate the primary logic of the problem. They include the most common and obvious cases that an average candidate would consider while coding. They do not include those cases that need extra checks to be placed in the logic.

Advanced: The advanced test-cases contain pathological input conditions that would attempt to break the codes which have incorrect/semi-correct implementations of the correct logic or incorrect/semi-correct formulation of the logic.

Edge: The edge test-cases specifically confirm whether the code runs successfully even under extreme conditions of the domain of inputs and that all possible cases are covered by the code

4 | Learning Resources

English Comprehension

[Improve your hold on the language by reading Shakespearan plays](#)



[Learn about how to get better at reading](#)



[Read opinions to improve your comprehension](#)



Logical Ability

[Practice Sherlock Holmes' puzzles and develop your deductive logic](#)



[Practice your Inductive Reasoning Skills!](#)



[Learn about generalizing unknown trends](#)



Quantitative Ability

[Learn about the principles of statistics](#)



[Learn about the real life applications of probability](#)



[Learn about the real life applications of logarithms](#)



Icon Index



Free Tutorial



Paid Tutorial



Youtube Video



Web Source



Wikipedia



Text Tutorial



Video Tutorial



Google Playstore