Sample Candidate

Test ID: 171113834779284 | 9876543210 | sample@email.com

Test Date: September 18, 2022

Automata Fix

86 /100

Automata Fix

86 / 100

Logical Error	Code Reuse	Syntactical Error
100 / 100	50 / 100	100 / 100

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 **Proctoring** section captures the output of the different proctoring features used during the test.

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

Automata Fix

86 / 100

This test measures the candidate's debugging skills. It checks her/his ability to fix logical and syntactical errors and to reuse an existing code.

The candidate is able to detect errors in logic and use/fix pre-written libraries of source codes to achieve the required functionality. Being fluent in reading and understanding source codes is a critical skill needed to take part in any advanced software engineering project.

3 | Response

Automata Fix

Code Replay 86 / 100

Question 1 (Language: Java)

The function/method ***reverseHalfArray*** accepts two arguments - *size,* an integer representing the size of the list and *inputList,* representing an input list of integers, respectively.
The function/method ***reverseHalfArray*** returns the elements of the reversed input list in-place from the second half.

For example, if the *inputList* is {20, 30, 10, 40, 50}, the function/method is expected to return {20, 30, 50,40,10}.

The function/method compiles successfully but fails to return the desired result for some test cases. Your task is to debug the code so that it passes all the test cases.

Original Code	Test cases passed: 33.33%	Response	Test cases passed: 100 %
3	void reverseHalfArray(int size, int inputList[])	3	void reverseHalfArray(int size, int inputList[])
4	{	4	{
5	int i, temp;	5	int i, temp;
6	for(i=0; i< size/2 ; i++)	6	for(i=size/2; i< size ; i++)
7	{	7	{
8	temp = inputList[size-1];	8	temp = inputList[size-1];
9	inputList[size-1] = inputList[i];	9	inputList[size-1] = inputList[i];

No change | Code insertions | Code deletions | Code edits | Skipped comment part

Compilation Statistics

3	3
Total attempts	Successful

Response time:	**00:01:12**
Average test case pass percentage per compile:	**33.3%**

Question 2 (Language: Java)

You are a software developer at XYZ Technologies. You are supposed to enhance the functionality of a particular module that controls a functionality based on the time of logging.

Given two time stamps, your task is to write a function/method that calculates the difference between them. The difference (which shall always be either a positive number or zero) would then be used by another module to perform checks on logging.

Developers at XYZ Technologies already use a predefined structure **Time** containing *hour*, *minute*, and *second* as members. A collection of functions/methods for performing some common operations on times is also available. You must make use of these functions/methods to calculate and return the difference.

You must complete the function/method ***difference_in_times*** accepts two arguments - *time1*, and *time2*, representing two **Times** and returns the difference in the number of seconds.

(Please refer to the Helper Code tab for details regarding the structure and functions/methods).

Original Code	Test cases passed: 0%	Response	Test cases passed: 0 %
4	int difference_in_times(Time time1, Time time2)	4	int difference_in_times(Time time1, Time time2)
5	{	5	{
6	// write your code here	6	// write your code here
		7	
7	}	8	}
8	}	9	}

☐ No change ☐ Code insertions ☐ Code deletions ☐ Code edits ☐ Skipped comment part

Compilation Statistics

0 Total attempts	0 Successful		
		Response time:	**00:01:59**
		Average test case pass percentage per compile:	**0%**

Question 3 (Language: Java)

The function/method ***manchester*** print a list with the following property: for each element in the input array *arr*, a counter is incremented if the bit *arr*[i] is the same as *arr*[i-1]. Then the increment counter value is added to the output array. If the bit *arr*[i] and *arr*[i-1] are different, then 0 is added to the output array. For the first bit in the input array, assume its previous bit to be 0. This encoding is stored and returned in a new array. For example, if *arr* is {0,1,0,0,1,1,1,0}, the method should return an array {1, 0, 0, 2, 0, 3, 4, 0}.

The function/method ***manchester*** accepts two argument *size*, an integer representing the length of the input array; *arr*, a list of integers representing an input array. Each element of *arr* represents a bit, 0 or 1.

The function/method compiles successfully but fails to return the desired result for some test cases due to incorrect implementation of the function/method ***manchester.*** Your task is to debug the code so that it passes all the test cases.

Original Code	Test cases passed: 0%	Response	Test cases passed: 100 %
1	public class Solution	1	public class Solution
2	{	2	{
3	void manchester(int len, int[] arr)	3	void manchester(int size, int[] arr)
4	{	4	{
5	int res[] = new int[len];	5	int res[] = new int[size];
6	boolean result;	6	boolean result;
7	int count =0;	7	int count =0;
8	for(int i = 0; i < len; i++)	8	for(int i = 0; i < size; i++)
9	{	9	{
10	if(i==0)	10	if(i==0)
11	result= (arr[i]==0);	11	result= (arr[i]==0);
12	else	12	else
13	result = (arr[i]==arr[i-1]);	13	result = (arr[i]==arr[i-1]);
14	res[i] = (result)?(0):(++count);	14	res[i] = (result)?(++count):(0);
15		15	
16	}	16	}
17	for(int i=0; i<len; i++)	17	for(int i=0; i<size; i++)
18	{	18	{
19	System.out.print(res[i]+" ");	19	System.out.print(res[i]+" ");
20	}	20	}

☐ No change ☐ Code insertions ☐ Code deletions ☐ Code edits ☐ Skipped comment part

Compilation Statistics

Total attempts	Successful
2	2

Response time: **00:01:32**

Average test case pass percentage per compile: **50%**

Question 4 (Language: Java)

You are given a predefined structure Point and also a collection of related functions/methods that can be used to perform some basic operations on the structure.

You must implement the function/method ***isTriangle*** which accepts three points *P1, P2, P3* as inputs and checks whether the given three points form the vertices of a triangle.

If they form a triangle, the function/method returns 1. Otherwise, it returns 0.

(Please refer to the Helper Code tab for details regarding the structure Point and the predefined functions/methods around it).

Original Code — **Test cases passed: 0%**

```
4   int isTriangle(Point P1, Point P2, Point P3)
5   {
6       // write your code here
7   }
8
9   }
```

Response — **Test cases passed: 100 %**

```
4   int isTriangle(Point P1, Point P2, Point P3)
5   {
6       // write your code here
7       double a,b,c;
8       a= P1.calculateDistance(P2);
9       b= P1.calculateDistance(P3);
10      c= P2.calculateDistance(P3);
11      if((a+b>c) && (a+c>b) && (b+c>a))
12          return 1;
13      else
14          return 0;
15  }
16
17  }
```

No change | Code insertions | Code deletions | Code edits | Skipped comment part

Compilation Statistics

Total attempts	Successful
1	1

Response time: 00:03:30

Average test case pass percentage per compile: 100%

Question 5 (Language: Java)

The function/method ***removeKLargestElement*** prints space-separated integers that are left after removing the K^{th} largest integer from the input list *arr*. If the given *K* is out of bounds (*K* > *len*) or K^{th} largest integer does not exist, then the function/method should print space-separated integers of the input list *arr*.
The function/method ***removeKLargestElement*** accepts three arguments - *len*, an integer representing the number of elements in the list, *K*, an integer value and *arr*, a list of integers.

The function/method ***removeKLargestElement*** uses another error-free function/method ***kLargestElement*** which accepts three arguments - *len,* an integer representing the number of elements in the list, *K*, an integer value and *arr*, a list of integers and returns an integer representing the K^{th} largest integer in the input list *arr*.

The function/method ***removeKLargestElement*** compiles unsuccessfully due to syntactical error. Your task is to debug the code so that it passes all the test cases.

Assumptions:
The inputs *len* and *K* are always non-negative integers.

Original Code — Test cases passed: 0%

```
39          }
40          for(i=pos;i<len-1;i++)
41          {
42              arr[i]=arr[i+1]:
43          }
44          int rarr1[] = new int[len-1];
45          for(i=0;i<len-1;i++)
```

Response — Test cases passed: 100 %

```
39          }
40          for(i=pos;i<len-1;i++)
41          {
42              arr[i]=arr[i+1];
43          }
44          int rarr1[] = new int[len-1];
45          for(i=0;i<len-1;i++)
```

No change | Code insertions | Code deletions | Code edits | Skipped comment part

Compilation Statistics

2	1
Total attempts	Successful

Response time:	00:00:49
Average test case pass percentage per compile:	50%

Question 6 (Language: Java)

The function/method ***printFibonacci*** accepts an integer *num,* representing a number.
The function/method ***printFibonacci*** returns the first *num* numbers of the Fibonacci series.
For example, given input 5, the function should print the string “0 1 1 2 3” (without quotes).

The function/method compiles successfully but fails to return the desired result for some test cases**.** Your task is to debug the code so that it passes all the test cases.

Original Code — **Test cases passed: 16.67%**

```
4   {
5       long num1 = 0;
6       long num2 = 1;
7       for (int i = 1; i < num; ++i)
8       {
9           System.out.print(num1+" ");
10          long sum = num1 + num2;
11          num2 = sum;
12          num1 = num2;
13      }
14  }
15 }
```

Response — **Test cases passed: 100 %**

```
4   {
5       long num1 = 0;
6       long num2 = 1;
7       for (int i = 1; i <= num; ++i)
8       {
9           System.out.print(num1+" ");
10          long sum = num1 + num2;
11          num1 = num2;
12          num2 = sum;
14      }
15  }
16 }
```

No change | Code insertions | Code deletions | Code edits | Skipped comment part

Compilation Statistics

2	2
Total attempts	Successful

Response time:	00:01:15
Average test case pass percentage per compile:	50%

Question 7 (Language: Java)

The function/method ***printPattern*** accepts an argument *num,* an integer.
The function/method ***printPattern*** returns the first *num* lines of the pattern as shown below.
For example, if *num* = 3, the pattern should be:
1 1
2 2 2 2
3 3 3 3 3 3

The function/method compiles successfully but fails to return the desired result for some test cases**.** Your task is to debug the code so that it passes all the test cases.

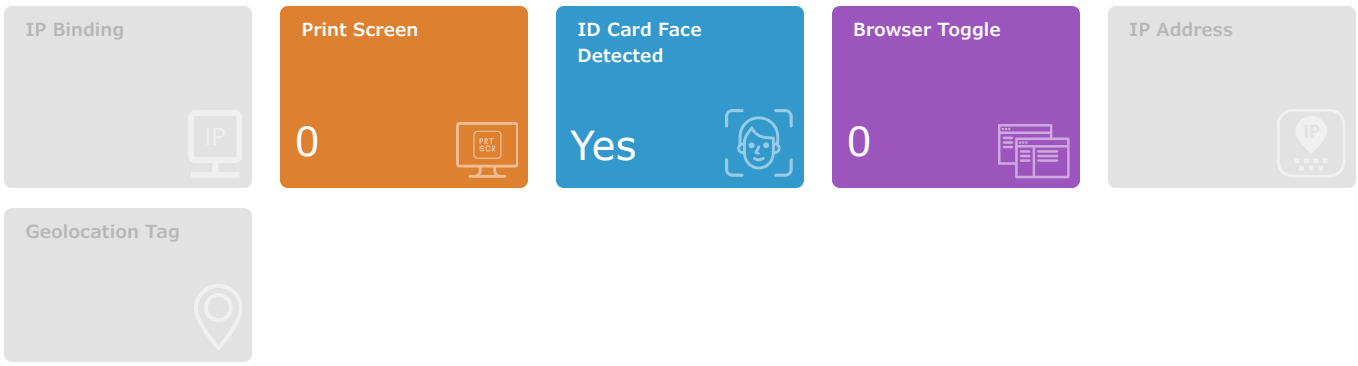
Original Code	Test cases passed: 16.67%	Response	Test cases passed: 100 %
5	int i,j;	5	int i,j;
6	for(i=1;i<=num;i++)	6	for(i=1;i<=num;i++)
7	{	7	{
8	for(j=i;j<=2*i;j++)	8	for(j=1;j<=2*i;j++)
9	{	9	{
10	System.out.print(j+" ");	10	System.out.print(i+" ");
11	}	11	}
12	System.out.println();	12	System.out.println();
13	}	13	}

No change | Code insertions | Code deletions | Code edits | Skipped comment part

Compilation Statistics

4 Total attempts	Response time: **00:01:23**
4 Successful	Average test case pass percentage per compile: **25%**

4 | Proctoring



AI Proctoring Information

Print Screen: The number of times the candidate attempted to take a screenshot of the assessment screen using the “print screen” function on their device. Note: This impacts proctoring index.

ID Card Face Detected: Looks at the candidate images captured during the assessment and flags anywhere different people appear to be present. Snapshots are included in the report.

Browser Toggle: Either the proportion of time the candidate spent focused on a tab/window other than that of assessment screen (%), or the number of times the candidate toggled to another tab/window (count). Note: This impacts proctoring index.

IP Address: Confirms that the candidate took the assessment from the specified IP address(s).

Geolocation Tag: Detects whether the candidate attempted the assessment from a location beyond the distance set by the administrator.