[290821] Set-Three-Web Dev-Fundamental Certification Exam

HIMANSHU | 29 Aug 2021

Finish State: **Normal**   Test Taken on: **August 29, 2021 11:56:10 AM IST**

H   HIMANSHU
        himanshu5336234@gmail.com

📅 Date of birth: Aug 21, 2000

📞 Contact No: 9639806383

Last Name: Not Filled

Finish State: **Normal**   Test Taken on: **August 29, 2021 11:56:10 AM IST**

H   HIMANSHU
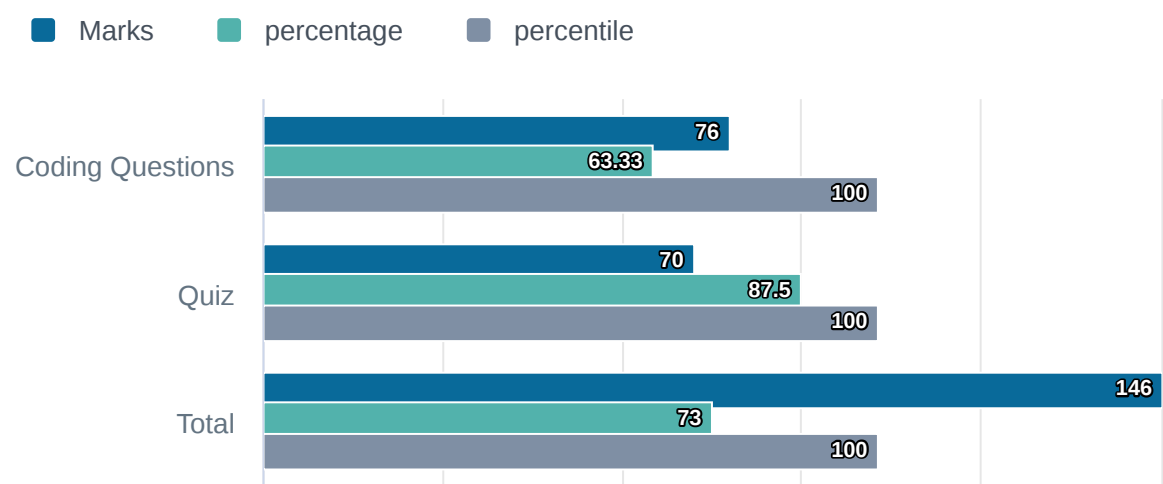   himanshu5336234@gmail.com

146 Marks Scored
out of 200

73 % 100 percentile
out of 5 Test Takers

2h 53m 28s Time taken
of 3hr

Performance Category: Passed

Marks Scored

■ Marks ■ percentage ■ percentile

Coding Questions
76
63.33
100

Quiz
70
87.5
100

Total
146
73
100

Attempt Summary

Distribution of questions attempted in a total of 43 question(s).

43
Total Questions

■ Correct	36 Ques	110/110 Marks
■ Incorrect	6 Ques	0/50 Marks
■ Partially Correct	1 Ques	36/40 Marks
■ Not Attempted	0 Ques	0/0 Marks

This shows the correctness of questions attempted by the test taker

Section-Wise Details

		question(s)	Time taken	Marks Scored
Section 1 Coding Questions		3 Q.	1h 22m 23s (Untimed)	76 / 120

Marks Scored

■ Marks ■ percentage

Category	Marks	percentage
Coding Skills - Linear…	40	100
Coding Skills - Basic …	36	90
Coding Skills - Basic …	0	0
Total	76	63.33
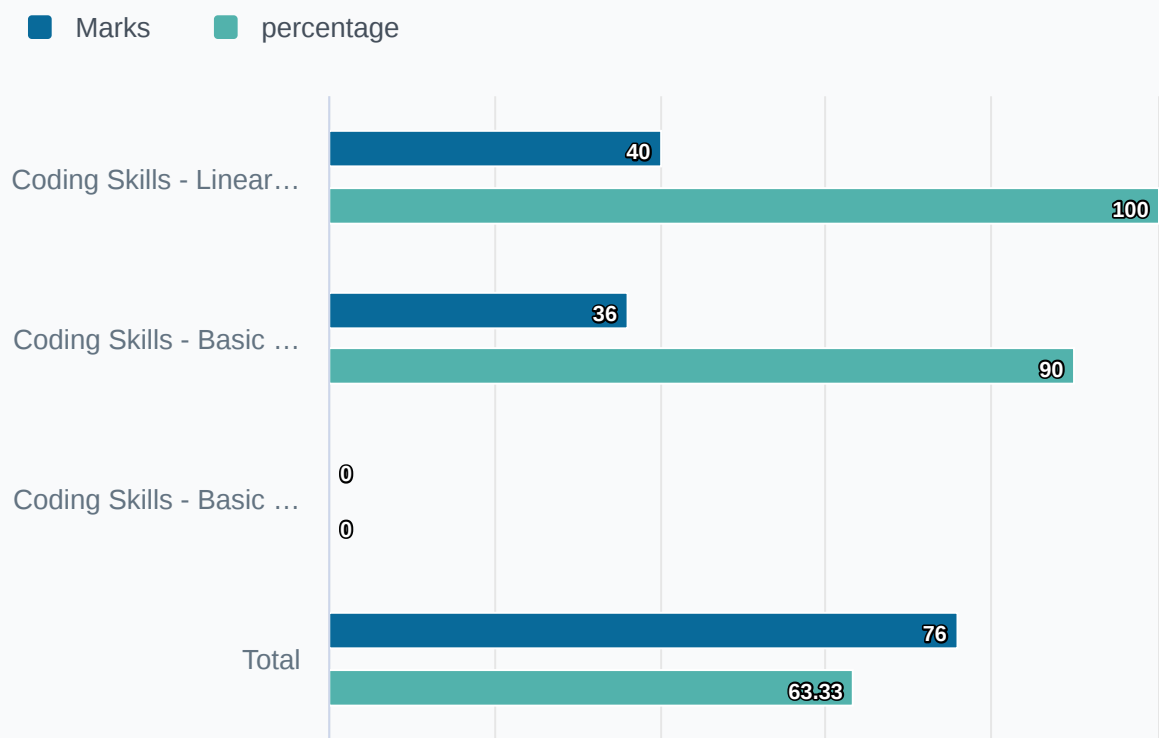
Attempt Summary

Distribution of questions attempted in a total of 3 question(s).

3
Total Questions
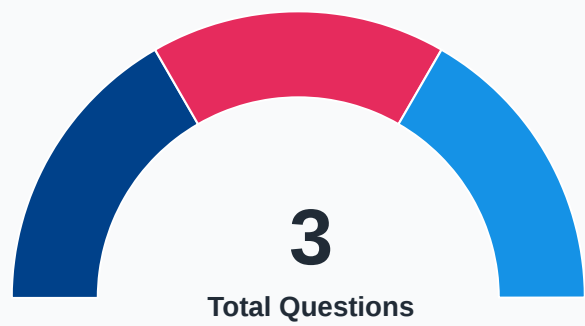
This shows the correctness of questions attempted by the test taker

■ Correct	1 Ques	40/40 Marks
■ Incorrect	1 Ques	0/40 Marks
■ Partially Correct	1 Ques	36/40 Marks

Section-Wise Details

		question(s)	Time taken	Marks Scored
▼	Section 2 **Quiz**	40 Q.	1h 31m 5s (Untimed)	70 / 80

Marks Scored

■ Marks ■ percentage

	Marks	percentage
EdYoda JavaScript M…	30	93.75
EdYoda HTML MCQs	20	83.33
EdYoda CSS MCQs	20	83.33
Total	70	87.5
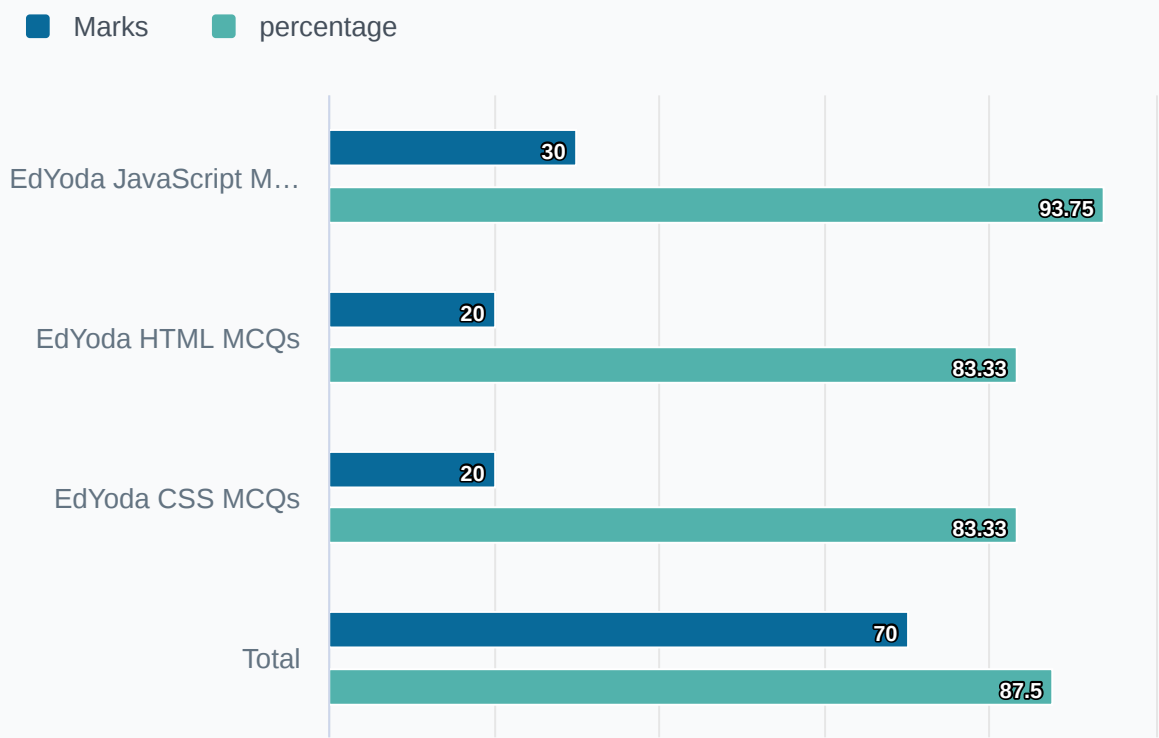
Attempt Summary

Distribution of questions attempted in a total of 40 question(s).

40
Total Questions
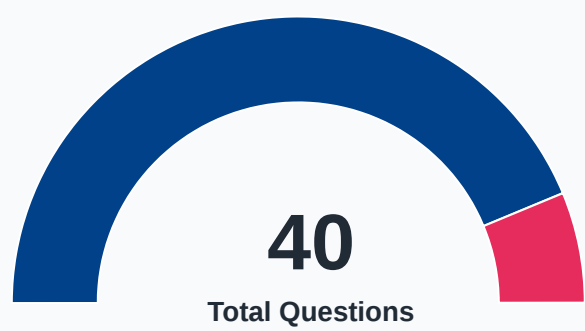
This shows the correctness of questions attempted by the test taker

■ Correct	35 Ques	70/70 Marks
■ Incorrect	5 Ques	0/10 Marks

Question-Wise Details

Coding Questions

	3	1h 22m 23s
	question(s)	Time taken

Q. 1

▼ Question 1

Compilation: **Successful**   🕐 Time taken: 37m 7s

Nick's Check

Nick has been given a list of random numbers by his teacher. These numbers are marks of several students of his class. He is required to arrange the marks in increasing order and hence check whether the new arrangement of marks are successive in nature or not. You need to write a function such that it returns 1 if the complete arrangement consists of consecutive marks, otherwise return 0.

Note: If two students have the same marks, then after arranging them in increasing order, they will not be considered as consecutive.

Input Specification:

input1: Integer N i.e., size of the array
input2: Integer array for elements of the array

Output Specification:

Return 1 if all the numbers are consecutive after arrangement, otherwise return 0.

Example 1:

input1: 6
input2: { 3,7,2,5,4,6}

Output: 1

Explanation:

After arranging the given numbers in increasing order, the array comes out to be as 2,3,4,5,6,7. As all the numbers in the array are consecutive, therefore 1 is returned.

Example 2:

input1: 6
input2: {1,9,2,4,0,3}

Output: 0

Explanation:

The given array, when arranged in increasing order, comes out to be 0,1,2,3,4,9. As all the numbers in the array are not consecutive, therefore 0 is returned.

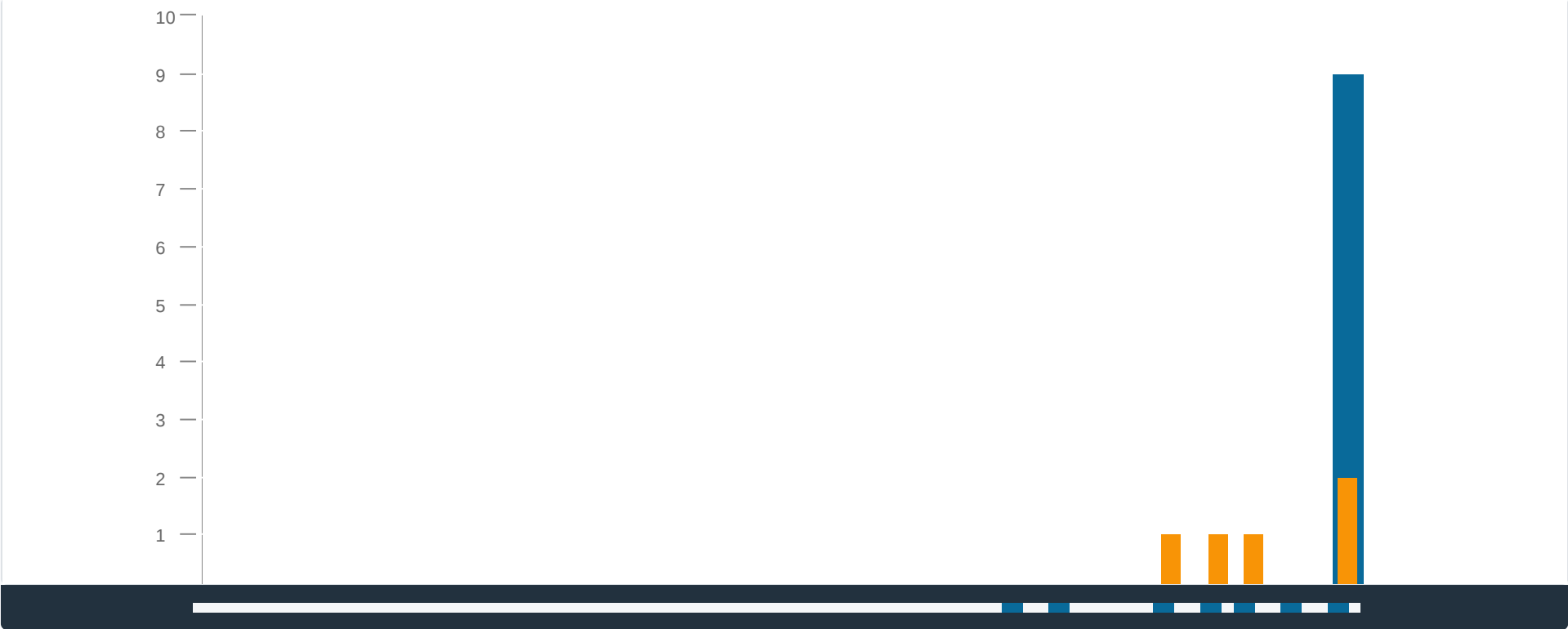EASY   Coding Skills - Basic - Array

Compilation	Time Taken	Language
✓ Successful	🕐 37m 7s	**JS** JAVASCRIPT8

```
1   // Read only region start
```
JAVASCRIPT8

```javascript
 2   function checkConsecutive(input1, input2) {
 3       // Read only region end
 4     input2.sort()
 5     for(var i=1;i<input1;i++){
 6         if(input2[i-1]+1!==input2[i]){
 7             return 0;
 8         }
 9     }
10
11     return 1;
12   }
```


Sample Test Case TimestampGraded Test Case Timestamp

Graded Test Case Code Compilation : 7 Successful 7 Attempt

Sample Test Case Code Compilation : 7 Successful 7 Attempt

Code complexity ⓘ : 0

Total no. of Testcase : 10   **Total Passed :** 9

TEST CASE	MARKS	CPU (MS)	PROCESSING (MS)	MEMORY (KB)	INPUTS	EXPECTED OUTPUT	ACTUAL OUTPUT	ERROR MESSAGE
Basic TestCase 1	4	0	342	104068	1, {1}	1	1	NA
Basic TestCase 2	0	0	358	104068	10, {1,2,4,3,5,9, 7,8,10,6}	1	0	NA
Basic TestCase 3	4	0	358	104324	20, {1,2,6,20,15, 16,8,13,10,1 1,3,14,5,17, 4,3,12,7,18, 19}	0	0	NA
Basic TestCase 4	4	0	388	104068	5, {2,1,4,7,6}	0	0	NA
Corner TestCase 1	4	0	413	104324	6, {1,2,3,4,5,6}	1	1	NA
Corner TestCase 2	4	0	361	104068	8, {8,7,6,5,4,3,	1	1	NA

					2,1}			
Necessary TestCase 1	4	0	352	104068	2, {2,0}	0	0	NA
Necessary TestCase 2	4	0	358	104324	4, {1,0,2,3}	1	1	NA
Time Complexity TestCase 1	4	0	382	104068	5, {1,4,5,6,7}	0	0	NA
Time Complexity TestCase 2	4	0	387	104324	7, {1,3,5,7,9,2, 4}	0	0	NA

Annual Day

On the eve of the annual day, a competition is held among **N** students in which each student gains some points based on their performance. All students were asked to stand in a queue in an increasing order based on the points they scored. The clever students stood in a random order so that the students with lower points are not noticed.

You will be given a points array where **points[i]** tells the teacher the number of points that the i th student earned. Your task is to help the teacher to find all pairs such that for all (0 <= i < j < N), the points scored by the i th student is greater than that of the j th student, i.e., **points[i] > points[j]** where i < j and return the total count of such pairs.

Input Specification:

input1: An integer value N (1 <= N <= 1000) denoting the number of students in the class
input2: An integer array of length N representing the points scored where input2[i] = points scored by i th student and input2[i] < = 1000 (0 <= i < N)

Output Specification:

Return the total number of pairs that satisfy the constraints given in the problem statement

Example 1:

input1: 10
input2: {1, 2, 7, 5, 7, 4, 1, 1, 2, 5}

Output: 20

Explanation:

The pairs of points satisfying the given scenario are :-

(2,1), (2,1), (7,5), (7,4), (7,1), (7,1), (7,2), (7,5), (5,4), (5,1), (5,1), (5,2), (7,4), (7,1), (7,1), (7,1), (7,2), (7,5), (4,1), (4,1), (4,2).

Hence, 20 is returned as the output.

Example 2:

input1: 5
input2: {1,1,3,6,2}

Output: 2

Explanation:

The pairs of points satisfying the given scenario are :-

(3,2) and (6,2).

Hence, 2 is returned as the output.

!

EASY	Coding Skills - Linear Search

	Compilation		Time Taken		Language
✓	Successful	🕐	3m 36s	JS	JAVASCRIPT8

```javascript
1   // Read only region start
2   function Student(input1, input2) {
3       // Read only region end
4       // Input parameters are :
5          // input1 As Number eg:- 23
6          // input2 As javascript Array() of type Number eg. var array = new Array()
7
8       // Expected return type  :
9          // output1 As Number eg:- 23
10      // Write code here...
11      var i=0;
12      var count =0;
13      while(i<input1){
14          var a =input2[i];
15          for(var j=i+1;j<input1;j++){
16              if(a>input2[j]){
17                  count++;
18              }
19          }
20          i++;
21      }
22      return count;
23      throw "UnsupportedException Student( input1, input2)";
24  }
```
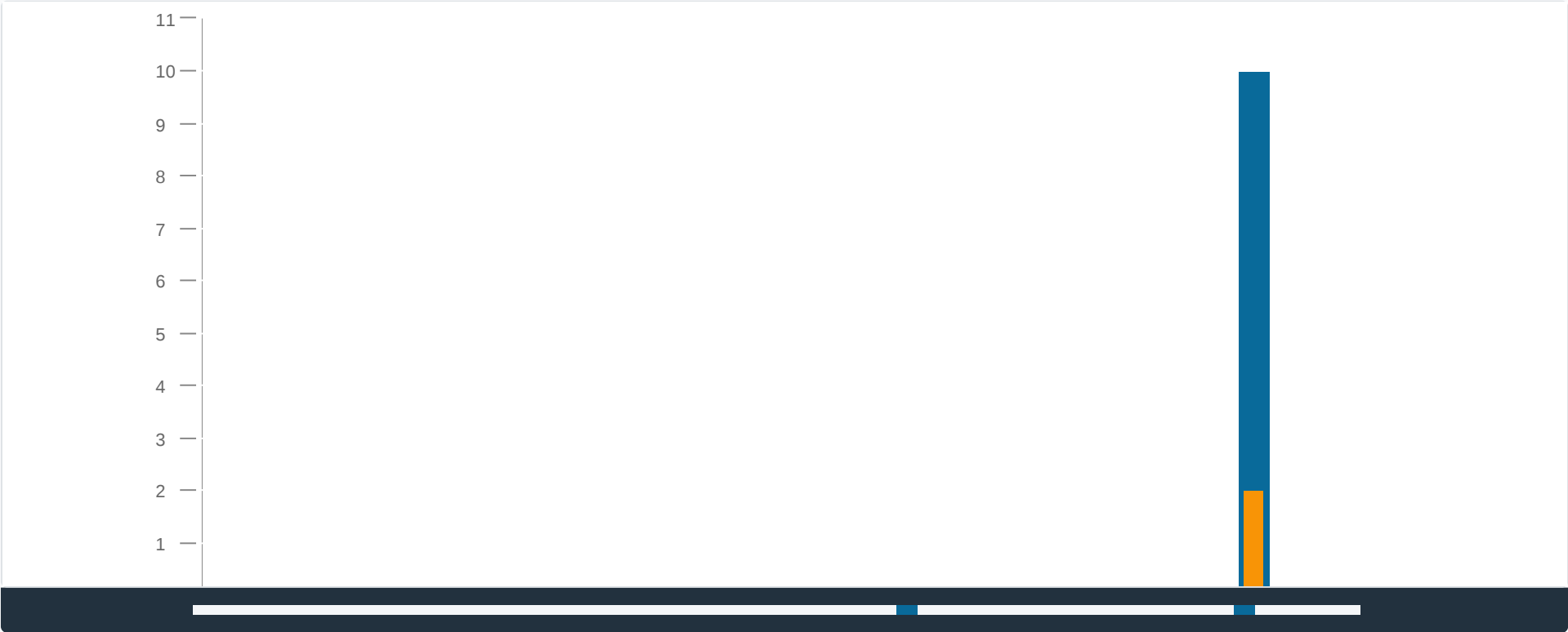

Sample Test Case Timestamp Graded Test Case Timestamp

Graded Test Case Code Compilation : **2** Successful **2** Attempt

Sample Test Case Code Compilation : **2** Successful **2** Attempt

Code complexity ⓘ : **0**

Total no. of Testcase : **10**   **Total Passed :** **10**

TEST CASE	MARKS	CPU (MS)	PROCESSING (MS)	MEMORY (KB)	INPUTS	EXPECTED OUTPUT	ACTUAL OUTPUT	ERROR MESSAGE
Corner Testcase 1	4	0	400	103812	140, {309, 27, 243, 292, 15, 91, 645, 826, 755, 171,	4705	4705	NA

					503, 479,			
					556, 709,			
					84, 151,			
					203, 141,			
					391, 761,			
					365, 59,			
					307, 280,			
					793, 375,			
					105, 643,			
					532, 615,			
					411, 560,			
					641, 653,			
					14, 374,			
					744, 377,			
					363, 380,			
					548, 584,			
					21, 822,			
					455, 104,			
					136, 376,			
					801, 245,			
					18, 47, 303,			
					43, 45, 814,			
					418, 705,			
					620, 112,			
					483, 193,			
					390, 5, 8,			
					403, 379,			
					470, 780,			
					460, 12,			
					209, 762,			
					33, 193, 98,			
					692, 47,			
					473, 655,			
					292, 490,			
					420, 594,			
					532, 464,			
					290, 668,			
					332, 72,			
					499, 814,			
					820, 51,			
					818, 827,			
					454, 78,			
					460, 115,			
					256, 190,			
					323, 180,			
					222, 515,			
					277, 77,			
					562, 468,			
					450, 572,			
					676, 33, 47,			
					90, 496,			
					336, 757,			
					546, 407,			
					418, 522,			
					389, 188,			
					222, 98,			
					641, 18,			
					276, 755,			
					274, 465,			

					796, 453, 406, 473, 449, 201, 753}			
Basic Testcase 3	4	0	416	104068	150, {25, 20, 9, 29, 18, 7, 25, 19, 19, 2, 5, 14, 5, 29, 21, 7, 18, 14, 26, 1, 12, 2, 6, 28, 14, 29, 16, 23, 6, 6, 23, 1, 17, 2, 29, 5, 8, 17, 23, 18, 18, 20, 24, 14, 19, 15, 21, 7, 20, 9, 28, 23, 3, 4, 21, 16, 24, 7, 9, 29, 4, 23, 22, 21, 24, 21, 25, 24, 29, 11, 12, 10, 1, 27, 23, 11, 13, 6, 9, 24, 15, 7, 18, 17, 10, 9, 3, 25, 8, 4, 24, 11, 26, 16, 23, 13, 29, 11, 7, 28, 21, 10, 29, 13, 8, 23, 23, 12, 28, 23, 6, 5, 29, 15, 21, 1, 16, 16, 17, 23, 11, 11, 25, 7, 19, 11, 11, 18, 21, 17, 8, 12, 19, 8, 16, 18, 1, 1, 29, 20, 23, 26, 25, 14, 12, 8, 6, 27, 23, 22}	5255	5255	NA
Time Complexity Testcase 1	4	0	503	104324	500, {129, 550, 530, 159, 413, 215, 530, 258, 448, 550, 621, 226, 336, 686, 600,	63096	63096	NA

					489, 773,			
					45, 177,			
					124, 319,			
					906, 913,			
					701, 373,			
					251, 799,			
					878, 664,			
					392, 585,			
					567, 574,			
					475, 503,			
					387, 666,			
					361, 789,			
					454, 926,			
					442, 196,			
					578, 796,			
					107, 859,			
					597, 622,			
					358, 390,			
					257, 901,			
					68, 64, 110,			
					727, 7, 846,			
					685, 345,			
					150, 511,			
					230, 432,			
					679, 961,			
					370, 482,			
					15, 759,			
					625, 894,			
					569, 227,			
					65, 489,			
					342, 868,			
					54, 842,			
					576, 763,			
					832, 653,			
					419, 77,			
					271, 634,			
					180, 612,			
					417, 885,			
					673, 415,			
					794, 71,			
					672, 803,			
					900, 628,			
					345, 142,			
					25, 143,			
					642, 734,			
					34, 844,			
					579, 958,			
					739, 1, 958,			
					462, 379,			
					23, 971, 4,			
					522, 910,			
					670, 949,			
					720, 101,			
					153, 414,			
					396, 377,			
					603, 195,			
					700, 229,			
					605, 200,			
					331, 378,			

					283, 280,			
					29, 651,			
					545, 766,			
					439, 705,			
					149, 649,			
					474, 668,			
					550, 198,			
					621, 892,			
					885, 853,			
					837, 877,			
					119, 61,			
					722, 428,			
					997, 169,			
					616, 623,			
					455, 904,			
					118, 100,			
					683, 420,			
					587, 46,			
					776, 628,			
					353, 229,			
					158, 496,			
					566, 746,			
					916, 977,			
					93, 369,			
					480, 583,			
					70, 736,			
					249, 476,			
					96, 82, 365,			
					161, 689,			
					905, 497,			
					531, 851,			
					851, 954,			
					252, 575,			
					780, 622,			
					849, 184,			
					155, 380,			
					439, 557,			
					26, 650,			
					372, 612,			
					778, 479,			
					403, 841,			
					541, 428,			
					750, 991,			
					982, 449,			
					741, 421,			
					123, 14,			
					583, 298,			
					969, 49,			
					337, 318,			
					701, 507,			
					173, 188,			
					320, 187,			
					318, 947,			
					490, 733,			
					699, 624,			
					216, 159,			
					110, 512,			
					374, 223,			
					544, 921,			

| | | | | | 98, 643, 622, 173, 202, 227, 739, 972, 839, 288, 45, 865, 535, 1, 771, 255, 282, 590, 705, 563, 351, 233, 257, 863, 90, 34, 105, 695, 938, 5, 815, 550, 771, 451, 642, 634, 162, 905, 643, 696, 497, 515, 472, 599, 825, 280, 928, 430, 714, 583, 448, 820, 697, 153, 905, 95, 597, 729, 322, 800, 294, 824, 631, 438, 542, 351, 781, 89, 502, 764, 376, 177, 165, 545, 127, 624, 8, 403, 357, 920, 880, 533, 790, 868, 575, 908, 80, 103, 588, 520, 700, 233, 840, 680, 620, 40, 861, 599, 153, 146, 213, 325, 851, 15, 848, 703, 643, 374, 724, 38, 204, 831, 559, 186, 978, 182, 646, 866, 44, 439, 230, 783, | | | | |

| | | | | | 580, 719, 477, 242, 718, 103, 103, 565, 137, 142, 628, 918, 423, 107, 347, 294, 534, 874, 850, 991, 722, 288, 448, 395, 400, 822, 62, 330, 631, 82, 340, 16, 39, 586, 578, 857, 95, 403, 421, 590, 118, 8, 444, 557, 410, 245, 634, 395, 838, 881, 8, 290, 600, 446, 163, 396, 436, 457, 53, 13, 118, 898, 658, 348, 681, 750, 609, 822, 949, 476, 30, 796, 882, 135, 834, 925, 566, 247, 345, 797, 605, 344, 164, 493, 95, 620, 373, 928, 873, 204, 603, 572, 870, 525, 841, 759, 127, 723, 651, 240, 597, 152, 785, 152, 89, 375, 329, 658, 218, 954, 160, 72, 146, 69, 379, 978, 160, 25, 667, 451, 771, 777} | | | |

| Necessary Testcase 2 | 4 | 0 | 398 | 104324 | 200, {16, 18, 63, 50, 1, 7, 13, 29, 7, 42, 49, 57, 65, 18, 32, 12, 31, 5, 26, 9, 59, 59, 18, 28, 31, 13, 66, 33, 14, 14, 33, 2, 16, 21, 62, 2, 31, 7, 11, 11, 26, 12, 28, 30, 53, 42, 41, 43, 10, 48, 18, 27, 40, 62, 33, 64, 24, 9, 19, 5, 46, 55, 14, 63, 59, 10, 20, 48, 53, 54, 24, 46, 47, 67, 14, 51, 46, 18, 6, 55, 21, 12, 4, 12, 52, 28, 2, 44, 35, 36, 15, 9, 7, 56, 61, 29, 17, 4, 59, 36, 20, 54, 32, 4, 66, 56, 25, 62, 25, 61, 58, 24, 1, 45, 54, 27, 17, 44, 10, 65, 34, 5, 67, 32, 66, 26, 24, 22, 29, 53, 12, 33, 63, 17, 22, 27, 29, 26, 59, 36, 40, 3, 49, 59, 52, 15, 20, 28, 20, 46, 7, 36, 2, 53, 19, 34, 62, 21, 35, 12, 2, 3, 67, 5, 45, 59, 46, 45, 31, 40, 13, 35, 36, 50, 40, 2, 67, 23, 41, 54, 14, | 9504 | 9504 | NA |

					3, 13, 42, 9, 31, 16, 52, 12, 2, 16, 27, 45, 50, 59, 12, 63, 23, 16, 38}			
Necessary Testcase 1	4	0	394	104068	200, {65, 104, 126, 100, 12, 170, 41, 55, 145, 114, 172, 126, 17, 191, 85, 76, 182, 78, 66, 119, 101, 85, 106, 172, 171, 171, 192, 138, 57, 158, 183, 101, 166, 183, 142, 133, 122, 103, 149, 151, 78, 111, 65, 38, 104, 159, 59, 61, 126, 74, 127, 106, 87, 43, 114, 45, 12, 73, 98, 111, 196, 3, 166, 146, 126, 6, 111, 44, 71, 56, 142, 107, 31, 125, 12, 146, 60, 80, 47, 144, 98, 157, 87, 180, 198, 143, 153, 54, 90, 54, 182, 154, 32, 163, 47, 157, 31, 80, 198, 29, 95, 10, 103, 105, 22, 70, 102, 78, 105, 186, 104, 187, 81, 12, 60, 29, 70, 109, 89, 171, 187, 69, 152, 183,	9966	9966	NA

					183, 86, 194, 117, 27, 196, 3, 132, 164, 28, 144, 40, 195, 72, 192, 171, 58, 129, 118, 96, 185, 68, 191, 184, 3, 74, 80, 150, 83, 150, 9, 96, 131, 156, 80, 56, 58, 159, 147, 177, 109, 119, 8, 54, 97, 136, 31, 145, 129, 169, 72, 158, 187, 99, 122, 29, 24, 120, 51, 54, 17, 10, 181, 96, 20, 43, 197, 48, 198, 116, 89, 167, 154, 169, 65, 199}			
Basic Testcase 2	4	0	418	102916	150, {36, 35, 42, 60, 80, 52, 62, 36, 101, 20, 44, 11, 13, 44, 99, 74, 92, 79, 39, 98, 87, 86, 63, 66, 42, 60, 29, 18, 76, 75, 45, 96, 76, 81, 57, 93, 91, 1, 101, 24, 87, 29, 38, 96, 55, 34, 56, 79, 17, 68, 60, 63, 36, 7, 22, 57, 17, 52, 50, 101, 90, 90, 89, 75, 5, 98, 98, 44, 101, 27, 25, 95, 41, 84, 17, 12, 78, 20, 73, 59,	6243		6243	NA

					26, 24, 16, 51, 85, 31, 50, 92, 4, 43, 66, 50, 81, 45, 35, 54, 18, 92, 44, 56, 32, 78, 35, 46, 70, 41, 102, 71, 88, 18, 29, 70, 90, 61, 45, 23, 93, 4, 6, 7, 10, 18, 40, 63, 41, 25, 70, 29, 97, 65, 74, 59, 34, 72, 56, 60, 11, 5, 9, 91, 31, 3, 7, 75, 14, 12, 30, 97, 9, 84}			
Basic Testcase 1	4	0	385	104068	50, {13, 4, 29, 12, 21, 20, 22, 11, 6, 6, 12, 31, 1, 8, 13, 23, 21, 5, 17, 1, 31, 14, 12, 5, 27, 2, 17, 29, 17, 14, 29, 10, 16, 6, 14, 5, 23, 26, 4, 24, 18, 2, 8, 28, 20, 19, 10, 14, 6, 26}	576	576	NA
Basic Testcase 4	4	0	374	104324	10, {1,2,5,6,4,3, 1,1,2,6}	19	19	NA
Time Complexity Testcase 2	4	0	449	104324	140, {1, 1, 1, 5, 5, 91, 645, 826, 755, 6, 503, 8, 556, 709, 84, 151, 203, 100, 391, 761, 345, 59, 307, 280, 793, 375, 105, 643, 532, 615, 411, 560, 641, 653, 14, 374,	4450	4450	NA

					744, 377, 363, 380, 548, 584, 21, 822, 345, 200, 136, 376, 801, 200, 18, 47, 303, 43, 45, 814, 418, 705, 620, 112, 483, 193, 390, 5, 8, 403, 379, 470, 780, 460, 12, 209, 762, 33, 193, 98, 692, 47, 473, 655, 292, 490, 420, 594, 532, 464, 290, 668, 332, 72, 499, 814, 820, 51, 818, 827, 454, 78, 460, 200, 256, 190, 323, 200, 222, 515, 277, 77, 562, 468, 450, 572, 676, 33, 47, 90, 496, 336, 757, 546, 407, 418, 522, 389, 188, 222, 98, 641, 18, 276, 755, 274, 465, 796, 453, 406, 473, 200, 201, 753}			
Corner Testcase 2	4	0	445	104068	340, {1, 1, 1, 5, 5, 91, 645, 826, 755, 6, 503, 8, 556, 709, 84, 151, 203, 100, 391, 761, 345, 59,	38997	38997	NA

					307, 280,			
					793, 375,			
					105, 643,			
					532, 615,			
					411, 560,			
					641, 653,			
					14, 374,			
					744, 377,			
					363, 380,			
					548, 584,			
					21, 822,			
					345, 200,			
					136, 376,			
					801, 200,			
					18, 47, 303,			
					43, 45, 814,			
					418, 705,			
					620, 112,			
					483, 193,			
					390, 5, 8,			
					403, 379,			
					470, 780,			
					460, 12,			
					209, 762,			
					33, 193, 98,			
					692, 47,			
					473, 655,			
					292, 490,			
					420, 594,			
					532, 464,			
					290, 668,			
					332, 72,			
					499, 814,			
					820, 51,			
					818, 827,			
					454, 78,			
					460, 200,			
					256, 190,			
					323, 200,			
					222, 515,			
					277, 77,			
					562, 468,			
					450, 572,			
					676, 33, 47,			
					90, 496,			
					336, 757,			
					546, 407,			
					418, 522,			
					389, 188,			
					222, 98,			
					641, 18,			
					276, 755,			
					274, 465,			
					796, 453,			
					406, 473,			
					200, 201,			
					753, 16, 18,			
					63, 50, 1, 7,			
					13, 29, 7,			

					42, 49, 57,				
					65, 18, 32,				
					12, 31, 5,				
					26, 9, 59,				
					59, 18, 28,				
					31, 13, 66,				
					33, 14, 14,				
					33, 2, 16,				
					21, 62, 2,				
					31, 7, 11,				
					11, 26, 12,				
					28, 30, 53,				
					42, 41, 43,				
					10, 48, 18,				
					27, 40, 62,				
					33, 64, 24,				
					9, 19, 5, 46,				
					55, 14, 63,				
					59, 10, 20,				
					48, 53, 54,				
					24, 46, 47,				
					67, 14, 51,				
					46, 18, 6,				
					55, 21, 12,				
					4, 12, 52,				
					28, 2, 44,				
					35, 36, 15,				
					9, 7, 56, 61,				
					29, 17, 4,				
					59, 36, 20,				
					54, 32, 4,				
					66, 56, 25,				
					62, 25, 61,				
					58, 24, 1,				
					45, 54, 27,				
					17, 44, 10,				
					65, 34, 5,				
					67, 32, 66,				
					26, 24, 22,				
					29, 53, 12,				
					33, 63, 17,				
					22, 27, 29,				
					26, 59, 36,				
					40, 3, 49,				
					59, 52, 15,				
					20, 28, 20,				
					46, 7, 36, 2,				
					53, 19, 34,				
					62, 21, 35,				
					12, 2, 3, 67,				
					5, 45, 59,				
					46, 45, 31,				
					40, 13, 35,				
					36, 50, 40,				
					2, 67, 23,				
					41, 54, 14,				
					3, 13, 42, 9,				
					31, 16, 52,				
					12, 2, 16,				

					27, 45, 50, 59, 12, 63, 23, 16, 38}			

Infix to Postfix Conversion

Infix Expressions:

In Infix expressions, operators are written in-between their operands.

An expression such as A * (B + C) / D means-

1. First add B and C together

2. Multiply the result by A

3. Divide by D to get the final answer

The expression for adding the numbers 1 and 2 is "1 + 2".

Postfix Expressions:

In Postfix expressions, the operators follow their operands.

The expression for adding the numbers 3 and 4 is "3 4 +".

If there are multiple operations, the operator is given immediately after its second operand. So the expression written for "3 − 4 + 5" would be "3 4 − 5 +" in Postfix notation. Here, 4 is first subtracted from 3, then added to 5.

Write a program that takes input as a string containing an infix expression and returns the corresponding postfix expression.

Note:

1. The string contains operators **(+,-,/,*),** parenthesis and operands **(digits)**.

2. Each digit is a separate operand.

The operator precedence is as follows:

1. **[**divide(/) and multiply(*) operator **] > [**addition(+) and subtraction(-) operator **]**.

2. If you encounter operators with same pecedence like (divide and multipy) or (addition and subtraction), then evaluate the operators from left to right.

Input Specification:

  input1: an input string that contains the Infix expression.

Output Specification:

  Return the Postfix expression of the given Infix expression.

Example 1:

  input1: 8+(7-9)*2

  Output: 879-2*+

Explanation:

=> 8+(7-9)*2

1. character: 8   stack: 8   operations: nil

2. character: +   stack: 8   operations: +

3. character: (stack: 8   operations: +(

4. character: 7   stack: 87   operations: +(

5. character: -   stack: 87   operations: +(-

6. character: 9   stack: 879   operations: +(-

7. character:)   stack: 879-   operations: +

8. character: *   stack: 879-   operations: +*

9. character: 2   stack: 879-2*+   operations: nil

Returned output: 879-2*+

Example 2:

input1: (1)+8

Output: 18+

Explanation:

=> (1)+8

1. character: (stack: nil   operations: (

2. character: 1   stack: 1   operations: (

3. character:)   stack: 1   operations: nil

4. character: +   stack: 1   operations: +

5. character: 8   stack: 18+   operations: nil

Returned output: 18+

!

MEDIUM   Coding Skills - Basic - Stack

	Compilation		Time Taken		Language
✓	Successful	🕐	41m 40s	JS	JAVASCRIPT8

```javascript
// Read only region start
function infixToPostfix(input1) {
    // Read only region end
    // Input parameters are :
        // input1 As String eg:- "abc"

    // Expected return type  :
        // output1 As String eg:- "abc"
    // Write code her
    var a=[];
    var b=[];
    for(var i=0;i<input1.length;i++){
        if(input1[i]=='+'||input1[i]=="*"||input1[i]=="/"||input1[i]=="("||input1[i]=="-"){
            b.push(input1[i]);
        }
        else if(input1[i]==")"){
            while(b[b.length-1]!='('){
                var c=b.pop();
                a.push(c);
            }
            b.pop();
        }
        else{
            a.push(input1[i]);
        }
    }
    while(b.length){
        var c=b.pop();
        a.push(c);
    }
    var postfix=""
 for(var j=0;j<a.length;j++){
        postfix +=a[j]
```

```
34    }
35    return postfix;
36    }
```
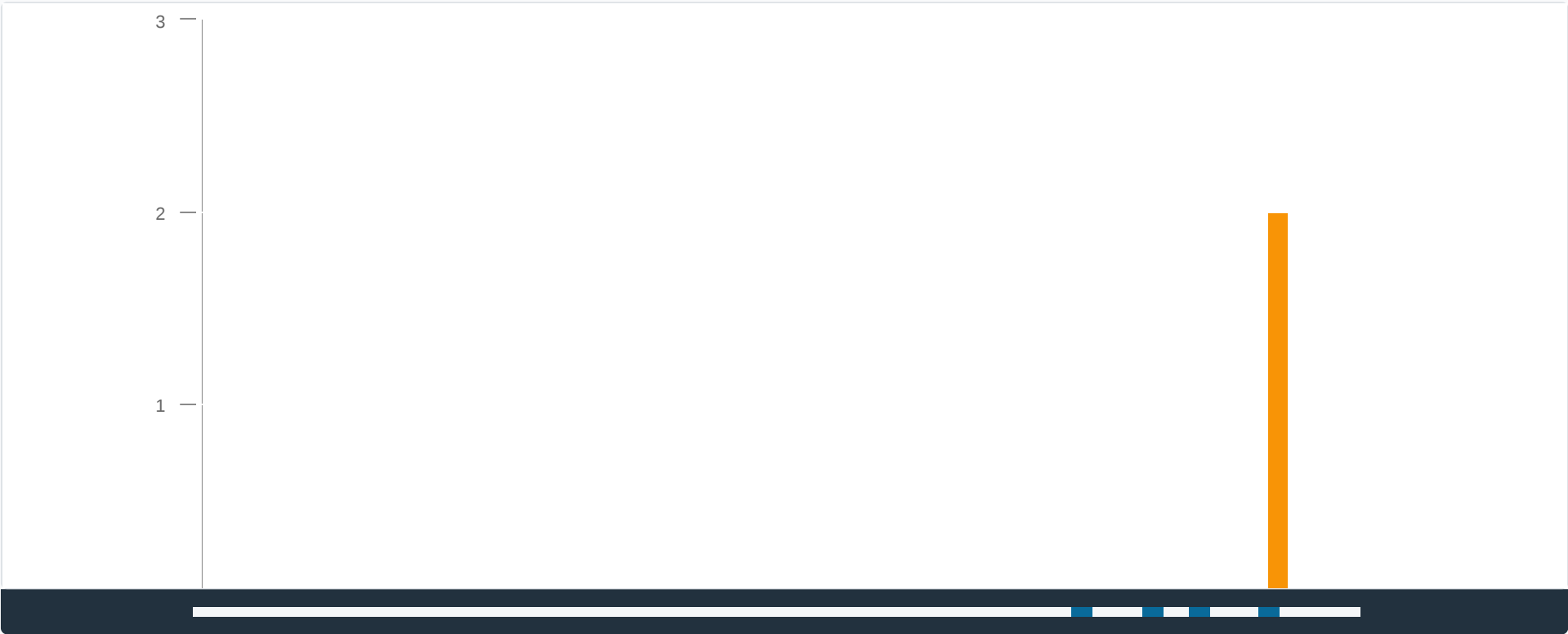
Sample Test Case Timestamp Graded Test Case Timestamp

Graded Test Case Code Compilation : 4 Successful   4 Attempt

Sample Test Case Code Compilation : 4 Successful   4 Attempt

Code complexity ⓘ : 0

Total no. of Testcase : 10   **Total Passed :** 0

TEST CASE	MARKS	CPU (MS)	PROCESSING (MS)	MEMORY (KB)	INPUTS	EXPECTED OUTPUT	ACTUAL OUTPUT	ERROR MESSAGE
Time Complexity Testcase 2	0	0	381	104068	((3*5-7*(((2)))+(8)/5+3-6/4+(7)+4*7-2*4/6+5/5/4*3*2*(6-4)+((((5))*((6)+9)+6-(2)*9/8+3+(1))*3+1)/(5)+((((9))))+(2*3+3+4)/((7))-(2-2)/(5)*(9)))	35*72*-85/+3+64/-7+47*+24*6/-55/4/3*2*64-*+569+*6+29*8/-3+1+3*1+5/+9+23*3+4+7/+22-5/9*-	3572853647 4724655432 64-569+629831 ++/*-+*31+*5923 34++*722-59*/-/++/+*** //+/*-*++/-+/+*-*	NA
Time Complexity Testcase 1	0	0	387	104324	(3+4-2*3-9)+2+2/4-(9)-(9*6-7-3-6-8+3*2+9/6*1+6-9-1/(3))*5/(2)*7/2-(((((6)))/(3)*7))/9/4-8*4/3+7+	34+23*-9-2+24/+9-96*7-3-6-8-32*+96/1*+6+9-13/-5*2/7*2/-63/7*9/4/-84*3/-7+18*8+94/-586+/5/-+	34239-*-+224996736 8329616913/--+*/+*+----*5272637*/9 4843718894 586+5//-/-+*++/*-//-/*/*--/++	NA

					((1)*8+8-9/4-(5)/(8+6)/5)			
Corner Testcase 2	0	0	403	104068	(1+4+9/(7)*3+3/(7)-5*8*7/4-(9-6*2/6+(4)))-(4/6+3/7/1)+((5))*4+3-3*3+6+5-8/((((7)-1))-9)*1/7/8-7*5*3+4	14+97/3*+37/+58*7*4/-962*6/-4+-46/37/1/+-54*+3+33*-6+5+871-9-/1*7/8/-75*3*-4+	1497337587496264+/*--/**-/+*/++46371//+/5433365871-9-1787534+**-//*/-++*-+*+-	NA
Corner Testcase 1	0	0	390	104324	(7)*4+(2)-(2)+8+8+((5+(8*5)))-(7)/8+1-((9/6*5))-7/(1+6/7)+7-(8)/(8)-(6)-(2-7)+(4)	74*2+2-8+8+585*++78/-1+96/5*-7167/+/-7+88/-6-27--4+	742288585*+781965*/7167/+788627-4+--/-+/--+/-+++-+*	NA
Necessary Testcase 2	0	0	400	104068	((((((8)*((7))+5+8-4*6+3*(8))+(((8)*4*9*(9)/(4)+7-5-6+9))))))	87*5+8+46*-38*+84*9*9*4/7+5-6-9++	87584638*+*-++*849947569+--+/***+	NA
Necessary Testcase 1	0	0	379	104068	(1/(2)+8+1-6)+(((((2)*7/3-4+4))))/(4*4+3)	12/8+1+6-27*3/4-4+44*3+/+	12816-++/27344+-/*443+*/+	NA
Basic Testcase 4	0	0	388	104068	((((8)+3-3)-(5)-7-7))	83+3-5-7-7-	833-+577---	NA
Basic Testcase 3	0	0	424	104324	(8)-6+7-1/5-8-1	86-7+15/-8-1-	8671581--/-+-	NA
Basic Testcase 2	0	0	407	104068	(9)*(1*8)/3	918**3/	918*3/*	NA
Basic Testcase 1	0	0	389	104068	(6-1+4)	61-4+	614+-	NA

Question-Wise Details

Quiz

40 question(s)	**1h 31m 5s** Time taken

Q. 1

▼ Question 1

🕐 Time taken: 21s

Before HTML5 where data had to be stored?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Cookies	✓	✓
Browser		
Google Chrome		
Internet Explorer		

Q. 2

▼ Question 2

🕐 Time taken: 19s

Which of the following is not a type of attribute for input tag?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
day	✓	✓
week		
month		
time		

▼ Question 3

What is the path for an image located in same folder as the current page?

EASY   EdYoda HTML MCQs

Response:

OPTIONS	RESPONSE	ANSWER
<img src= "pic.jpg">	✓	✓
<img src= "../pic.jpg">		
<img src= "images/pic.jpg">		
<img src= "/images/pic.jpg">		

▼ Question 4

Consider the following markup and answer the question that follow.

```
<source src="html_5.mp4" type="video/mp4">
<source src="html_5.ogv" type="video/ogg">
```

What is the need to add multiple file formats for the same file?

EASY   EdYoda HTML MCQs

Response:

OPTIONS	RESPONSE	ANSWER
To provide fallback support		
To address the media support problem		
To provide fallback support & address the media support problem	✓	✓
To provide foul back support		

Question 5

The link to a page on a different site is known as:

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Internal link		
External link	✓	✓
Intra-page link		
Targeted		

Question 6

Link bookmarks are identified with which of the following symbols?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
#	✓	✓
$		
%		
!		

Question 7

If you want to have the browser open a new window for a link to an outside page, which of the following attributes would you use?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
alt		
target	✓	✓
src		
start		

Question 8

Which of the following is the default style for the ordered list?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
upper-alpha		✓
decimal	✓	
numeric		
upper-roman		

Question 9

Which of the following tags creates a cell in the table?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
td	✓	✓
tr		
th		
caption		

Question 10

You want three cells in a row merged into one. Which of the following is the correct attribute and value in this case?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
rowspan="3"		
rowspan="2"	✓	
colspan="3"		✓
colspan="2"		

Question 11

Which of the following HTML statements will display an image in the web browser?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
		
		
	✓	✅
		

Question 12

Which of the following tags is not a valid HTML input type?

EASY   **EdYoda HTML MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
text		
checkbox		
radio		
textarea	✓	✅

Question 13

Time taken: 42s

align-items allows you to do what?

EASY **EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
align flex items along the main axis		
align flex items along the cross axis	✓	✓
align flex items horizontally		
align flex items vertically		

Question 14

Time taken: 52s

What describes block elements the best?

EASY **EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Block elements take only the width required by the content		
Block elements take the entire width of the webpage		
Block elements take the entire available width of the parent element	✓	✓
None of the above		

Question 15

What would the following code do on a screen of width 900 px?

<h1>Main Heading</h1>

@media(max-width: 400px) {

h1{

font-size: 24px;

}

}

@media(max-width: 800px) {

h1{

font-size: 48px;

}

}

@media(min-width: 1200px) {

h1{

font-size: 64px;

}

}

EASY   **EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
h1 tag will have a font-size of 48px		
h1 tag will have a font-size of 24px		
h1 tag will have a font-size of 64px	✓	
None of the above		✓

▼ Question 16

🕐 Time taken: 1m 56s

flex-direction allows you to do what?

EASY   **EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
define the direction of how elements are positioned based on either row or column	✓	✓
define how elements will wrap when the browser width is changed		
define how to position elements vertically		
defines how position elements horizontally		

▼ Question 17

🕐 Time taken: 4m 5s

Which of these is the limitation of inline elements

EASY   **EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Inline elements cannot have text in them		
Inline elements cannot have images		
Inline elements cannot have a button inside	✓	
Inline elements cannot have a custom width and height		✓

Question 18

Which of the following CSS3 code snippets can be used to place one background image on top of another?

EASY   EdYoda CSS MCQs

Response:

OPTIONS	RESPONSE	ANSWER

#background {
background-image: url('boy.jpg');
background-image: url('car.jpg');
}		

#background {
background-image: url('boy.jpg'), url('car.jpg');
}
	✓	✓

It is not possible		
None of the above		

Question 19

What is the font size of the paragraph in px?

<div style="font-size: 32px">
<p style="font-size: 1rem;">This is a para</p>
</div>

EASY   EdYoda CSS MCQs

Response:

OPTIONS	RESPONSE	ANSWER
16px	✓	✓
24px		
32px		
None of the above		

▼ Question 20

Time taken: 14s

How can you add a shadow to your element?

EASY　**EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
shadow: 2px 2px 2px #ccc;		
element-shadow: 2px 2px 2px #ccc;		
box-shadow: 2px 2px 2px #ccc;	✓	✓
shadow: 2px 2px 2px 2px 2px 2px #ccc;		

▼ Question 21

Time taken: 19s

How can you create a solid blue border with 10px thickness?

EASY　**EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
border: 10px solid blue;	✓	✓
element-border: 10px solid blue;		
element-outline: 10px solid blue;		
None of the above		

▼ Question 22

Nikhil wants to create a block which stays in position so that when he scrolls his page down it stays on top of the screen and does not affect anything in its environment. How will he do it?

EASY   **EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER

div {
width: 800px;
position: absolute;
top: 0;
left:0;
}		

div {
width: 800px;
position: fixed;
top: 0;
left:0;
}
	✓	✓

div {
width: 800px;
position: static;
top: 0;
left:0;
}		

div {
width: 800px;
position: relative;
top: 0;
left:0;
}		

▼ Question 22

Nikhil wants to create a block which stays in position so that when he scrolls his page down it stays on top of the screen and does not affect anything in its environment. How will he do it?

EASY   **EdYoda CSS MCQs**

Question 23

What is the use of "background-size: contain" property in CSS3?

EASY   **EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
It scales the background image so that the content area is completely covered by the background image		
It scales the background image to be as large as possible keeping the width and the height inside the content area	✓	✓
It scales the background image so that the content area is completely covered by the width of the background image		
It scales the background image so that the content area is completely covered by the height of the background image		

Choose the correct section names in the sequence from Section-1 to Section-4.

EASY **EdYoda CSS MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Border, Margin, Padding, Content		
Margin, Padding, Border, Content		
Margin, Border, Padding, Content	✓	✓
None of the above		

Question 24

▼ Question 25

What will be printed in the console?

console.log(parseInt(false))
console.log(Number(false))

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
0 and 0		
NaN and NaN		
NaN and 0	✓	✓
Error		

▼ Question 26

What will be printed in the console?

```
var num1 = 10;
var num2 = 20;
const num3 = 30;
function mNum() {
    var num2 = 40;
    let num3 = 50;
    console.log(num1 + num2 + num3);
}
mNum();
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
60		
70		
100	✓	✓
Error		

Question 27

What is the observation made in the following JavaScript code?

```
var count = [1, , 3];
```

EASY **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
The omitted value takes "undefined"	✓	✓
This results in an error		
This results in an exception		
The omitted value takes an integer value		

What will be printed in the console?

```javascript
class Person {
    constructor() {
        this.name = 'John';
        this.age = 30;
    }
    getName() {
        return this.name;
    }
}

class Athelete extends Person {
    constructor() {
        super();
        this.occupation = 'Shooter';
    }
    getDetails() {
        console.log(this.getName() + ' works as a ' + this.occupation);
    }
}

const mObj = new Athelete();
mObj.getDetails();
console.log(mObj.getName());
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
"John works as a Shooter" and "John"	✓	✓
Error		
"John works as a Shooter" and Error		
"John works as a Shooter"		

Question 29

What will be printed in the console?

var num1 = 100;

var result = num1 > 50 ? (num1+10) < 100 ? (num1 + 20) : (num1 + 50) < 200 ? (num1 + 500) : (num1 + 200) : num1;
console.log(result)

EASY   EdYoda JavaScript MCQs

Response:

OPTIONS	RESPONSE	ANSWER
120		
150		
100		
600	✓	✓

Question 30

What will be printed in the console?

```javascript
const mArr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
mArr.slice(0, 5);
mArr.splice(2, 3);
console.log(mArr);
```

EASY   EdYoda JavaScript MCQs

Response:

OPTIONS	RESPONSE	ANSWER
[1, 2, 6, 7, 8, 9, 10]	✓	✓
[1, 2]		
Error		
[1, 2, 6]		

What will the following function do?

```javascript
const mFunc = (arg) => {
    if (arg !== undefined && arg !== null) {
        return Object.keys(arg).map((objKey) => {
            return (
                [...Array(arg[objKey])].map((orderItem, key) => {
                    return ({
                        id: arg[objKey].id,
                        name: arg[objKey].name
                    });
                })
            )
        }).reduce((acc, item) => {
            return acc.concat(item)
        }, []);
    } else {
        return []
    }
}
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Takes a JSON array as an input and returns another JSON array		
Take a JSON object as an input and returns a JSON array	✓	✓
Takes a JSON object as an input and returns a JSON object		
Takes a JSON array as an input and return a JSON object		

Question 32

What will be printed in the console?

console.log('This is another /'test/' string')

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
This is another /'test/' string		
This is another test string		
This is another 'test' string		
Error	✓	✓

Question 33

What will be the result of running the following code?

```
const mObj = {
    firstName: 'John',
    lastName: 'Lark',
    age: 28,
    getDetails: function () {
        console.log(`${firstName} ${lastName} is ${age} years old`);
    }
}
mObj.firstName = 'Toto'
mObj.lastName = 'Wolff';
mObj.age = 35;
mObj.getDetails();
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
John Lark is 28 years old		
Toto Wolff is 28 years old		
Toto Wolff is 35 years old	✓	
None of the above		✓

Question 34

What will be printed in the console?

```
const mObj = {
    "name": 'Baba Yaga',
    "getName": function () {
        console.log(this.name)
    }
}
console.log(JSON.stringify(mObj))
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
{"name":"Baba Yaga", "getName":"function () {console.log(this.name)}"}		
Error		
{"name":"Baba Yaga"}	✓	✓
None of the above		

Question 35

What will be printed in the console?

```javascript
const mObj = {
    "name": 'Baba Yaga',
    "getName": function () {
        console.log(this.name)
    }
}

localStorage.setItem('user-details', mObj);
console.log(localStorage.getItem('user-details'));
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
{"name":"Baba Yaga", "getName":"function () {console.log(this.name)}"}		
{"name":"Baba Yaga", "getName":[function Function]}		
{"name":"Baba Yaga"}		
[object Object]	✓	✓

Question 36

What is the equivalent jquery function for the following JavaScript function?

```javascript
function mFunc(selectorStr, func) {
    var lis = document.querySelectorAll(selectorStr);
    for (i = 0; i < lis.length; i++) {
        lis[i].onclick = func;
    }
}
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
keydown()		
keyup()		
click()	✓	✓
dblclick()		

Question 37

What will be printed in the console?

```
function printDetails(n) {
    return function (a) {
        return function (o) {
            console.log(`${n} is ${a} years old and works as a ${o}`);
        }
    }
}
printDetails('Toto')(38)('Team Principal');
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Error: Undefined function declaration		
Toto is 38 years old and works as a Team Principal	✓	✓
Runtime Error		
Team Principal is 38 years old and works as a Toto		

Question 38

What does the location property represent?

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Current DOM object		
Current URL	✓	✓
Both DOM object and URL		
Document		

What will be printed in the console?

```javascript
const mArr = [1, 2, 3, 4, 5];
const mStr = 'Amigo';
const mNum1 = 10;
const mNum2 = 20;
const mNum3 = 30;
const result = mNum1 + mNum2 + mStr + mArr + mNum3;
console.log(result);
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
Error		
1020Amigo1,2,3,4,530		
1020Amigo1,1020Amigo121020Amigo1,1020Amigo131020Amigo1,1020Amigo14,1020Amigo1530		
30Amigo1,2,3,4,530	✓	✓

▼ Question 40

What will be printed in the console?

```javascript
var mNum = 10;
const mProm1 = new Promise((resolve, reject) => {
    if (mNum >= 10) {
        resolve(mNum);
    } else {
        reject(new Error('Number not big enough'));
    }
})

const mProm2 = (data) => new Promise((resolve, reject) => {
    data = data + 20;
    if (data >= 30) {
        resolve(data);
    } else {
        reject(new Error('Data not big enough'));
    }
})

mProm1
    .then(mProm2)
    .then((res) => {
        console.log(res);
    })
    .then(mProm2)
    .then(response => console.log(response))
    .catch((err) => {
        console.log(err);
    })
```

EASY   **EdYoda JavaScript MCQs**

Response:

OPTIONS	RESPONSE	ANSWER
30		
50		
Error		
30 and Error	✓	✓

Test Log

29th Aug 2021

09:02 AM	🏳	Started the test with Quiz
09:42 AM	●	Went to Coding Questions of the test
09:47 AM	●	Went to Quiz of the test
09:55 AM	●	Went to Coding Questions of the test
09:57 AM	●	Went to Quiz of the test
09:58 AM	●	Went to Coding Questions of the test
10:08 AM	●	Went to Quiz of the test
10:29 AM	●	Went to Coding Questions of the test
10:30 AM	●	Went to Quiz of the test
10:31 AM	●	Went to Coding Questions of the test
11:21 AM	●	Went to Quiz of the test
11:28 AM	●	Went to Coding Questions of the test
11:41 AM	●	Went to Quiz of the test
11:44 AM	●	Went to Coding Questions of the test
11:48 AM	●	Went to Quiz of the test
11:56 AM	🏁	Finished the test