

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <any>
4 #include <vector>
5
6 using namespace std;
7
8 // O(n) time | O(d) space - where n is the total number of elements in
9 // array, including sub-elements, and d is the greatest depth of "spec
10 // arrays in the array
11 int productSum(vector<any> array, int multiplier = 1) {
12     int sum = 0;
13     for (auto el : array) {
14         if (el.type() == typeid(vector<any>)) {
15             sum += productSum(any_cast<vector<any>>(el), multiplier + 1);
16         } else {
17             sum += any_cast<int>(el);
18         }
19     }
20     return sum * multiplier;
21 }
22
```

Solution 1Solution 2Solution 3

```
1 #include <any>
2 #include <vector>
3
4 using namespace std;
5
6 int productSum(vector<any> array) {
7     // Write your code here.
8     return -1;
9 }
10
```

Our Tests

Custom Output

Submit Code

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int main() {
7     // Test Case 1: [1, 2, 3]
8     // Expected: 1 * 2 * 3 = 6
9     // Actual: 6
10     // Test Case 2: [1, 2, 3, 4]
11     // Expected: 1 * 2 * 3 * 4 = 24
12     // Actual: 24
13     // Test Case 3: [1, 2, 3, 4, 5]
14     // Expected: 1 * 2 * 3 * 4 * 5 = 120
15     // Actual: 120
16     // Test Case 4: [1, 2, 3, 4, 5, 6]
17     // Expected: 1 * 2 * 3 * 4 * 5 * 6 = 720
18     // Actual: 720
19     // Test Case 5: [1, 2, 3, 4, 5, 6, 7]
20     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 = 5040
21     // Actual: 5040
22     // Test Case 6: [1, 2, 3, 4, 5, 6, 7, 8]
23     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 = 40320
24     // Actual: 40320
25     // Test Case 7: [1, 2, 3, 4, 5, 6, 7, 8, 9]
26     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 = 362880
27     // Actual: 362880
28     // Test Case 8: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
29     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 = 3628800
30     // Actual: 3628800
31     // Test Case 9: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
32     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 = 39916800
33     // Actual: 39916800
34     // Test Case 10: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
35     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 = 479001600
36     // Actual: 479001600
37     // Test Case 11: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
38     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 = 6229516800
39     // Actual: 6229516800
40     // Test Case 12: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
41     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 = 87178291200
42     // Actual: 87178291200
43     // Test Case 13: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
44     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 = 1307674368000
45     // Actual: 1307674368000
46     // Test Case 14: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
47     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 = 20901888000000
48     // Actual: 20901888000000
49     // Test Case 15: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
50     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 = 355685824000000
51     // Actual: 355685824000000
52     // Test Case 16: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]
53     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 = 6341492400000000
54     // Actual: 6341492400000000
55     // Test Case 17: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
56     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 = 121669156800000000
57     // Actual: 121669156800000000
58     // Test Case 18: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
59     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 = 2432986304000000000
60     // Actual: 2432986304000000000
61     // Test Case 19: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]
62     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 = 51090963264000000000
63     // Actual: 51090963264000000000
64     // Test Case 20: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]
65     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 = 1124000640000000000000
66     // Actual: 1124000640000000000000
67     // Test Case 21: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]
68     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 = 25852614400000000000000
69     // Actual: 25852614400000000000000
70     // Test Case 22: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
71     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 = 620448384000000000000000
72     // Actual: 620448384000000000000000
73     // Test Case 23: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
74     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 = 15511210240000000000000000
75     // Actual: 15511210240000000000000000
76     // Test Case 24: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]
77     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 = 403291466240000000000000000
78     // Actual: 403291466240000000000000000
79     // Test Case 25: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]
80     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 = 10986869587200000000000000000
81     // Actual: 10986869587200000000000000000
82     // Test Case 26: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
83     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 = 307032348416000000000000000000
84     // Actual: 307032348416000000000000000000
85     // Test Case 27: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]
86     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 = 8504904055680000000000000000000
87     // Actual: 8504904055680000000000000000000
88     // Test Case 28: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
89     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 = 224462161440000000000000000000000
90     // Actual: 224462161440000000000000000000000
91     // Test Case 29: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]
92     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 = 5997171166080000000000000000000000
93     // Actual: 5997171166080000000000000000000000
94     // Test Case 30: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]
95     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 = 162598477337600000000000000000000000
96     // Actual: 162598477337600000000000000000000000
97     // Test Case 31: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]
98     // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 = 4287058465849600000000000000000000000
99     // Actual: 4287058465849600000000000000000000000
100    // Test Case 32: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]
101    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 = 115585527722892800000000000000000000000
102    // Actual: 115585527722892800000000000000000000000
103    // Test Case 33: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]
104    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 = 2939638218637248000000000000000000000000
105    // Actual: 2939638218637248000000000000000000000000
106    // Test Case 34: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36]
107    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 = 74426650091043840000000000000000000000000
108    // Actual: 74426650091043840000000000000000000000000
109    // Test Case 35: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37]
110    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 = 1906308452257141760000000000000000000000000
111    // Actual: 1906308452257141760000000000000000000000000
112    // Test Case 36: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]
113    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 = 48563819758287392000000000000000000000000000
114    // Actual: 48563819758287392000000000000000000000000000
115    // Test Case 37: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
116    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 * 39 = 1229596799949209600000000000000000000000000000
117    // Actual: 1229596799949209600000000000000000000000000000
118    // Test Case 38: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]
119    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 * 39 * 40 = 31744919998726400000000000000000000000000000000
120    // Actual: 31744919998726400000000000000000000000000000000
121    // Test Case 39: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41]
122    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 * 39 * 40 * 41 = 804772319967974400000000000000000000000000000000
123    // Actual: 804772319967974400000000000000000000000000000000
124    // Test Case 40: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42]
125    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 * 39 * 40 * 41 * 42 = 20520179239237056000000000000000000000000000000000
126    // Actual: 20520179239237056000000000000000000000000000000000
127    // Test Case 41: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]
128    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 * 39 * 40 * 41 * 42 * 43 = 522504554003283200000000000000000000000000000000000
129    // Actual: 522504554003283200000000000000000000000000000000000
130    // Test Case 42: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44]
131    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 * 39 * 40 * 41 * 42 * 43 * 44 = 13180113296074304000000000000000000000000000000000000
132    // Actual: 13180113296074304000000000000000000000000000000000000
133    // Test Case 43: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45]
134    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 * 39 * 40 * 41 * 42 * 43 * 44 * 45 = 332002829901806720000000000000000000000000000000000000
135    // Actual: 332002829901806720000000000000000000000000000000000000
136    // Test Case 44: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46]
137    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 29 * 30 * 31 * 32 * 33 * 34 * 35 * 36 * 37 * 38 * 39 * 40 * 41 * 42 * 43 * 44 * 45 * 46 = 8410071837244710400000000000000000000000000000000000000
138    // Actual: 8410071837244710400000000000000000000000000000000000000
139    // Test Case 45: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47]
140    // Expected: 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 * 14 * 15 * 16 * 17 * 18 * 19 * 20 * 21 * 22 * 23 * 24 * 25 * 26 * 27 * 28 * 2
```

```
1  vector<int> test = {25, 15, 10, 5, 0};
2  vector<int> test2 = {25};
3
4
5
6  //Test Case 2
7  vector<int> test = {25, 15, vector<int>{25, 15, 10},
8
9
10     vector<int>{25}};
11
12
13
14  //Test Case 3
15  vector<int> test = {vector<int>{25, 15, 10}, vector<int>{25, 10},
16     vector<int>{25}};
17
18
19
```

Run or submit code when you're ready.