

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4
5 using namespace std;
6
7 class LinkedList {
8 public:
9     int value;
10    LinkedList *next;
11
12    LinkedList(int value) {
13        this->value = value;
14        this->next = NULL;
15    }
16 };
17
18 // O(n + m) time | O(1) space - where n is the number of nodes in the
19 // Linked List and m is the number of nodes in the second Linked List
20 LinkedList *mergeLinkedLists(LinkedList *headOne, LinkedList *headTwo)
21 {
22     LinkedList *p1 = headOne;
23     LinkedList *p1Prev = NULL;
24     LinkedList *p2 = headTwo;
25     while (p1 != NULL && p2 != NULL) {
26         if (p1->value < p2->value) {
27             p1Prev = p1;
28             p1 = p1->next;
29         } else {
30             if (p1Prev != NULL)
31                 p1Prev->next = p2;
32             p1Prev = p2;
33             p2 = p2->next;
34             p1Prev->next = p1;
35         }
36     }
37     if (p1 != NULL)
38         p1Prev->next = p1;
39     if (p2 != NULL)
40         p1Prev->next = p2;
41     return p1Prev;
42 }
```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2
3 using namespace std;
4
5 // This is an input class. Do not edit.
6 class LinkedList {
7 public:
8     int value;
9     LinkedList *next;
10
11    LinkedList(int value) {
12        this->value = value;
13        next = NULL;
14    }
15 };
16
17 LinkedList *mergeLinkedLists(LinkedList *headOne, LinkedList *headTwo)
18 {
19     // Write your code here.
20     return NULL;
21 }
```

Our Tests

Custom Output

Submit Code

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

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

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

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

