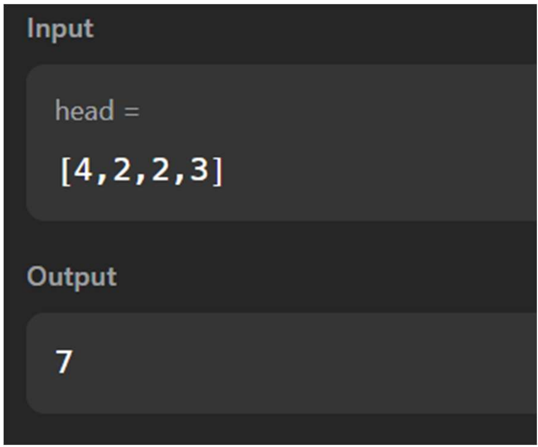
```
#include <stdio.h>
#include <stdlib.h>

// Definition for singly-linked list.

// Function to find the maximum twin sum
int pairSum(struct ListNode* head) {
    if (head == NULL || head->next == NULL) {
        // Empty or single-node list
        return 0;
    }

    // Traverse the linked list and find the twin sum
    int maxSum = 0;
    struct ListNode* current = head;
    int n = 0;

    // Count the number of nodes in the linked list
    while (current != NULL) {
        n++;
        current = current->next;
    }

    // Reset current to the head of the list
    current = head;

    // Traverse the first half of the linked list
    for (int i = 0; i < n / 2; i++) {
        int twinIndex = n - 1 - i;

        // Find the twin node
        struct ListNode* twin = head;
        for (int j = 0; j < twinIndex; j++) {
            twin = twin->next;
        }

        // Calculate the twin sum
        int twinSum = current->val + twin->val;

        // Update the maximum twin sum
        if (twinSum > maxSum) {
            maxSum = twinSum;
        }

        // Move to the next pair of nodes
        current = current->next;
    }

    return maxSum;
}
```



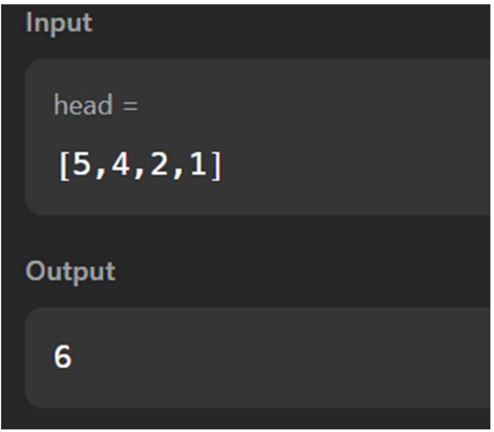
Input

head =

```
[4,2,2,3]
```

Output

```
7
```



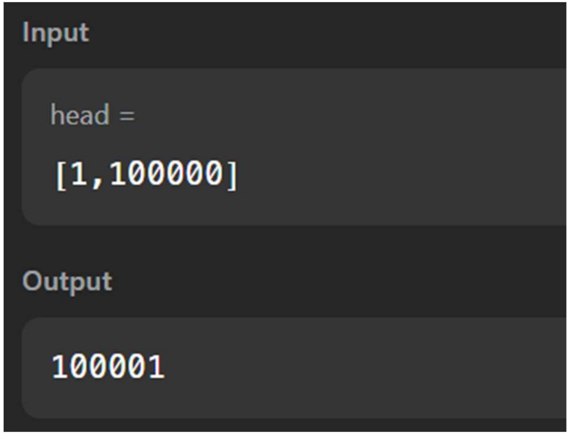
Input

head =

```
[5,4,2,1]
```

Output

```
6
```



Input

head =

```
[1,100000]
```

Output

```
100001
```