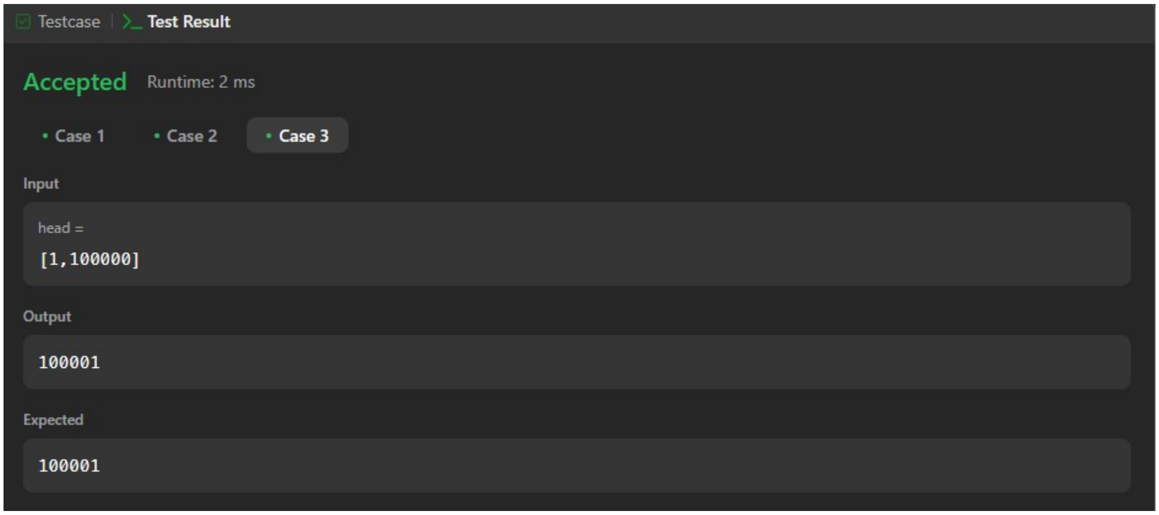
```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *      int val;
 *      struct ListNode *next;
 * };
 */
 struct ListNode* reverse(struct ListNode* head)
{
        struct ListNode*p = NULL, *q = NULL, *r = head;

        while(r!= NULL){
            p = q;
            q = r;
            r = r->next;
            q->next = p;
        }
        head = q;
        return head;
}

int pairSum(struct ListNode* head) {
    if(head == NULL){
            return -1;
}


//if only 2 nodes
if(head->next->next ==NULL){
    int sum = head->val + head->next->val;
    return sum;
}


struct ListNode *temp = head ,*s = head , *f = head->next;


// find middle
while(f!=NULL){
    f = f->next;
    if(f!=NULL){
        f = f->next;
        s = s->next;
    }
}


struct ListNode* second = reverse(s->next);
s->next = second;
struct  ListNode* first = head;


int ans = INT_MIN;
```

```
        while(second != NULL){
            int data = first->val + second->val;
            ans = fmax(ans,data);
            first = first->next;
            second = second->next;
        }

        return ans;
}
```



Accepted Runtime: 2 ms

Input

head =

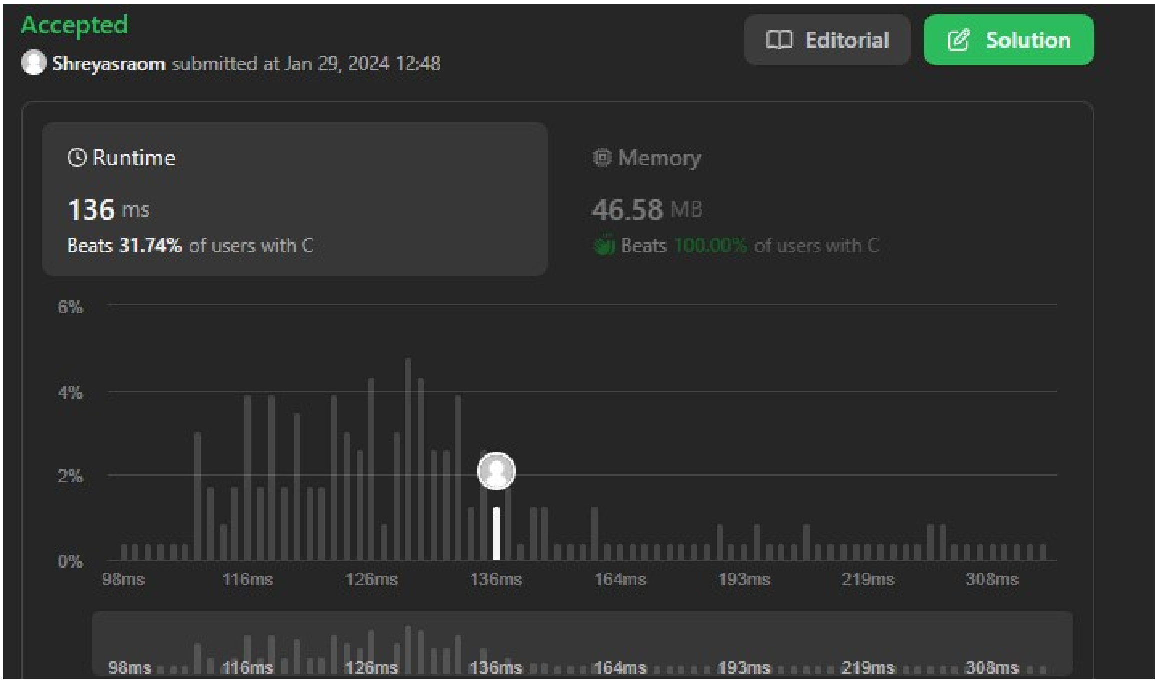
```
[1,100000]
```

Output

```
100001
```

Expected

```
100001
```



Accepted

Shreyasraom submitted at Jan 29, 2024 12:48

Runtime

136 ms

Beats 31.74% of users with C

Memory

46.58 MB

Beats 100.00% of users with C