**SOLVE (SET A)**

**Python**

| **def length(head):**  **count = 0**  **while head != None:**  **count += 1**  **head = head.next**  **return count**  **def getElem(head, idx):**  **i = 0**  **while(head != None):**  **if i == idx:**  **return head.elem**  **head = head.next**  **i += 1**  **def twoEndSum(head):**  **n = length(head)**  **ans = getElem(head, 0) + getElem(head, n - 1)**  **for i in range(1, n // 2):**  **ans -= (getElem(head, i) + getElem(head, n - 1 - i))**  **print(ans)** |
| --- |

**Java**

| **int length(Node head){**  **int count = 0;**  **while(head != null){**  **count += 1;**  **head = head.next;**  **}**  **return count;**  **}**  **int getElem(Node head, int idx){**  **int i = 0;**  **while(head != Null){**  **if(i == idx){**  **return head.elem;**  **}**  **head = head.next;**  **i += 1;**  **}**  **}**  **void twoEndSum(Node head) {**  **int n = length(head);**  **int ans = (getElem(head, 0) + getElem(head, n - 1));**  **for(int i = 1; i < n / 2; i += 1){**  **ans -= (getElem(head, i) + getElem(head, n - 1 - i));**  **}**  **print(ans);**  **}** |
| --- |

**SOLVE (SET B)**

**Python**

| **def length(head):**  **count = 0**  **while head != None:**  **count += 1**  **head = head.next**  **return count**  **def getElem(head, idx):**  **i = 0**  **while(head != None):**  **if i == idx:**  **return head.elem**  **head = head.next**  **i += 1**  **def twoEndSum(head):**  **n = length(head)**  **ans = getElem(head, 0) - getElem(head, n - 1)**  **for i in range(1, n // 2):**  **ans += (getElem(head, i) + getElem(head, n - 1 - i))**  **print(ans)** |
| --- |

**Java**

| **int length(Node head){**  **int count = 0;**  **while(head != null){**  **count += 1;**  **head = head.next;**  **}**  **return count;**  **}**  **int getElem(Node head, int idx){**  **int i = 0;**  **while(head != Null){**  **if(i == idx){**  **return head.elem;**  **}**  **head = head.next;**  **i += 1;**  **}**  **}**  **void twoEndSum(Node head) {**  **int n = length(head);**  **int ans = (getElem(head, 0) - getElem(head, n - 1));**  **for(int i = 1; i < n / 2; i += 1){**  **ans += (getElem(head, i) - getElem(head, n - 1 - i));**  **}**  **print(ans);**  **}** |
| --- |

**RUBRIC**

| **Category** | **Marks** |
| --- | --- |
| **Proper method/function definition with proper parameters** | **2** |
| **Properly Calculating the length of the linked list** | **2** |
| **Properly running the loop** | **3** |
| **Properly fetch value from desired nodes (getElem)** | **4** |
| **Calculating the values correctly** | **3** |
| **print the answer correctly** | **1** |
| **Note: There can be multiple solutions which produce the correct answer. Use your best judgement to evaluate the scripts.** | |