

IT DS 201 LAB

SUBMITTED BY ADITYA SINGH 2K19/EP/005

Program 15 : Given a Sorted doubly linked list, finds all the pairs that are equal to the given integer value.

CODE

```
void pairSum(Node *head, int x){
    Node *first = head;
    Node *second = head;
    while (second->next != NULL) second = second->next;

    bool found = false;

    while (first != second && second->next != first){
        if ((first->data + second->data) == x){
            found = true;
            cout << "(" << second->data << ", " << first->data << ") \n";

            first = first->next;
            second = second->prev;
        }
        else{
            if ((first->data + second->data) > x)
                first = first->next;
            else
                second = second->prev;
        }
    }
    if (found == false)
        cout << "No pair found";
}
```

```

struct Node{
    int data;
    Node *next, *prev;

    Node(int x){
        data = x; next = NULL; prev = NULL;
    }
};

void insert(Node **head, int data){
    Node *temp = new Node(data);
    temp->next = temp->prev = NULL;
    if (!(*head)) (*head) = temp;
    else{
        temp->next = *head;
        (*head)->prev = temp;
        (*head) = temp;
    }
}

```

ALGORITHM

1. Initialize first with the start of the doubly linked list and initialize second with the last node of the doubly linked list.
2. We initialize first and second pointers as first and last nodes. Here we don't have random access, so to find the second pointer, we traverse the list to initialize the second.
3. If the current sum of first and second is less than x, then we move first in forward direction. If the current sum of the first and second element is greater than x, then we move the second in the backward direction.
4. Loop termination conditions are also different from arrays. The loop terminates when two pointers cross each other, or they become the same.
5. The case when no pairs are present will be handled by the condition "first==second".

INPUT/OUTPUT

```
insert(&head, 9);  
insert(&head, 8);  
insert(&head, 5);  
insert(&head, 4);  
insert(&head, 3);  
insert(&head, 2);  
insert(&head, 1);
```

```
int x = 10;  
pairSum(head, x);
```

```
(1, 9)  
(2, 8)  
[Finished in 1.3s]
```

```
insert(&head, 2);  
insert(&head, 5);  
insert(&head, 7);  
insert(&head, 8);  
insert(&head, 9);  
insert(&head, 10);  
insert(&head, 12);  
insert(&head, 16);  
insert(&head, 19);  
insert(&head, 25);
```

```
int x = 35;  
pairSum(head, x);
```

```
(10, 25)  
(16, 19)  
[Finished in 1.3s]
```

END