Started on	Wednesday, 28 February 2024, 6:35 PM
State	Finished
Completed on	Wednesday, 28 February 2024, 7:27 PM
Time taken	51 mins 48 secs
Marks	20.00/20.00
Grade	**10.00** out of 10.00 (**100**%)

Question 1

Correct

Mark 10.00 out of 10.00

This challenge is part of a tutorial track by MyCodeSchool and is accompanied by a video lesson.

Given a pointer to the head of a singly-linked list, print each $data$ value from the reversed list. If the given list is empty, do not print anything.

Example

$head*$ refers to the linked list with $data$ values $1 \rightarrow 2 \rightarrow 3 \rightarrow NULL$

Print the following:

```
3
2
1
```

Function Description

Complete the *reversePrint* function in the editor below.

reversePrint has the following parameters:

- *SinglyLinkedListNode pointer head:* a reference to the head of the list

Prints

The $data$ values of each node in the reversed list.

Input Format

The first line of input contains t, the number of test cases.

The input of each test case is as follows:

- The first line contains an integer n, the number of elements in the list.
- Each of the next *n* lines contains a data element for a list node.

Constraints

- $1 \leq n \leq 1000$
- $1 \leq list[i] \leq 1000$, where $list[i]$ is the i^{th} element in the list.

Sample Input

```
3
5
16
12
4
2
5
3
7
3
9
5
5
1
18
3
13
```

Sample Output

```
5
2
4
12
16
9
3
7
13
3
18
1
5
```

Explanation

There are three test cases. There are no blank lines between test case output.

The first linked list has 5 elements: $16 \rightarrow 12 \rightarrow 4 \rightarrow 2 \rightarrow 5$. Printing this in reverse order produces:

```
5
2
4
12
16
```

The second linked list has 3 elements: $7 \rightarrow 3 \rightarrow 9 \rightarrow NULL$. Printing this in reverse order produces:

```
9
3
7
```

The third linked list has 5 elements: $5 \rightarrow 1 \rightarrow 18 \rightarrow 3 \rightarrow 13 \rightarrow NULL$. Printing this in reverse order produces:

```
13
3
18
1
5
```

For example:

Input	Result
3 5 16 12 4 2 5 3 7 3 9 5 5 1 18 3 13	5 2 4 12 16 9 3 7 13 3 18 1 5
3 3 11 1 17 3 12 11 15 4 5 7 15 14	17 1 11 15 11 12 14 15 7 5

Answer: (penalty regime: 0 %)

Reset answer

```
class SinglyLinkedList {
    public:
        SinglyLinkedListNode *head;
        SinglyLinkedListNode *tail;

        SinglyLinkedList() {
            this->head = nullptr;
            this->tail = nullptr;
        }

        void insert_node(int node_data) {
```

```
            SinglyLinkedListNode* node = new SinglyLinkedListNode(node_data);

            if (!this->head) {
                this->head = node;
            } else {
                this->tail->next = node;
            }

            this->tail = node;
        }
};

void print_singly_linked_list(SinglyLinkedListNode* node, string sep) {
    while (node) {
        cout << node->data;

        node = node->next;

        if (node) {
            cout << sep;
        }
    }
}

void free_singly_linked_list(SinglyLinkedListNode* node) {
    while (node) {
        SinglyLinkedListNode* temp = node;
        node = node->next;

        free(temp);
    }
}

/*
 * Complete the 'reversePrint' function below.
 *
 * The function accepts INTEGER_SINGLY_LINKED_LIST llist as parameter.
 */

/*
 * For your reference:
```

	Input	Expected	Got	
✔	3 5 16 12 4 2 5 3 7 3 9 5 5 1 18 3 13	5 2 4 12 16 9 3 7 13 3 18 1 5	5 2 4 12 16 9 3 7 13 3 18 1 5	✔

	Input	Expected	Got	
✔	3 3 11 1 17 3 12 11 15 4 5 7 15 14	17 1 11 15 11 12 14 15 7 5	17 1 11 15 11 12 14 15 7 5	✔

Passed all tests! ✔

▸ Show/hide question author's solution (Cpp)

Correct

Marks for this submission: 10.00/10.00.

Question 2

Correct

Mark 10.00 out of 10.00

Alexa has two stacks of non-negative integers, stack $a[n]$ and stack $b[m]$ where index 0 denotes the top of the stack. Alexa challenges Nick to play the following game:

- In each move, Nick can remove one integer from the top of either stack a or stack b.
- Nick keeps a running sum of the integers he removes from the two stacks.
- Nick is disqualified from the game if, at any point, his running sum becomes greater than some integer $maxSum$ given at the beginning of the game.
- Nick's *final score* is the total number of integers he has removed from the two stacks.

Given a, b, and $maxSum$ for g games, find the maximum possible score Nick can achieve.

Example

$a = [1, 2, 3, 4, 5]$
$b = [6, 7, 8, 9]$

The maximum number of values Nick can remove is 4. There are two sets of choices with this result.

1. Remove $1, 2, 3, 4$ from a with a sum of 10.
2. Remove $1, 2, 3$ from a and 6 from b with a sum of 12.

Function Description

Complete the *twoStacks* function in the editor below.

twoStacks has the following parameters: - *int maxSum:* the maximum allowed sum
- *int a[n]:* the first stack
- *int b[m]:* the second stack

Returns

- *int:* the maximum number of selections Nick can make

Input Format

The first line contains an integer, g (the number of games). The $3 \cdot g$ subsequent lines describe each game in the following format:

1. The first line contains three space-separated integers describing the respective values of n (the number of integers in stack a), m (the number of integers in stack b), and $maxSum$ (the number that the sum of the integers removed from the two stacks cannot exceed).
2. The second line contains n space-separated integers, the respective values of $a[i]$.
3. The third line contains m space-separated integers, the respective values of $b[i]$.

Constraints

- $1 \leq g \leq 50$
- $1 \leq n, m \leq 10^5$
- $0 \leq a[i], b[i] \leq 10^6$
- $1 \leq maxSum \leq 10^9$

Subtasks

- $1 \leq n, m, \leq 100$ for 50% of the maximum score.

Sample Input 0

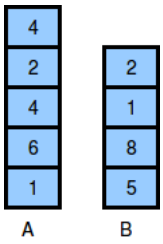
```
1
5 4 10
4 2 4 6 1
2 1 8 5
```

Sample Output 0

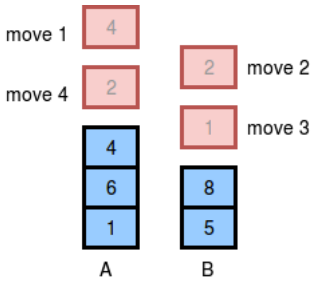
```
4
```

Explanation 0

The two stacks initially look like this:



The image below depicts the integers Nick should choose to remove from the stacks. We print $\mathbf{4}$ as our answer, because that is the maximum number of integers that can be removed from the two stacks without the sum exceeding $x = 10$.



(There can be multiple ways to remove the integers from the stack, the image shows just one of them.)

For example:

Input	Result
1 5 4 10 4 2 4 6 1 2 1 8 5	4
3 7 2 668 12 54 75 66 99 22 66 93 32 3 10 541 34 60 55 47 68 67 23 18 99 24 39 56 12 5 7 580 29 21 75 81 73 42 32 49 22 48 91 67	9 11 11

Answer: (penalty regime: 0 %)

Reset answer

```
#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);

/*
 * Complete the 'twoStacks' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 *  1. INTEGER maxSum
 *  2. INTEGER_ARRAY a
 *  3. INTEGER_ARRAY b
 */

int twoStacks(int maxSum, vector<int> a, vector<int> b) {
    int n = a.size();
    int m = b.size();

    // Initialize pointers and sum
    int sum = 0, count = 0;
    int i = 0, j = 0;

    // Move the pointer of the first stack until the sum exceeds maxSum
```

```
    while (i < n && sum + a[i] <= maxSum) {
        sum += a[i];
        i++;
    }

    count = i;

    // Start removing elements from the second stack and updating the count
    while (j < m && i >= 0) {
        sum += b[j];
        j++;

        // If the sum exceeds maxSum, remove elements from the first stack until it's within limit
        while (sum > maxSum && i > 0) {
            i--;
            sum -= a[i];
        }

        // Update the count if the current sum is valid and greater than the previous count
        if (sum <= maxSum && i + j > count) {
            count = i + j;
        }
    }

    return count;
```

	Input	Expected	Got	
✔	1 5 4 10 4 2 4 6 1 2 1 8 5	4	4	✔
✔	3 7 2 668 12 54 75 66 99 22 66 93 32 3 10 541 34 60 55 47 68 67 23 18 99 24 39 56 12 5 7 580 29 21 75 81 73 42 32 49 22 48 91 67	9 11 11	9 11 11	✔

Passed all tests! ✔

▸ Show/hide question author's solution (Cpp)

Correct

Marks for this submission: 10.00/10.00.