

Name: - Tirth Kothadiya
Enrollment No. : - 92410133008
Doman: - Microsoft

Easy: - Two Sum - Pair with Given Sum

Given an array arr[] of integers and another integer target. Determine if there exist two distinct indices such that the sum of their elements is equal to the target.

Code: -

```
class Solution {
public:
    bool twoSum(vector<int>& arr, int target) {
        unordered_set<int> seen;// hash set for Two Sum
        for (int num : arr) { //complement num = target
            if (seen.count(target - num)) return true;//number is seen than write true
            seen.insert(num);//element is not seen than insert into num
        }
        return false;//element sum not found
    }
};
```


Output: -

Output Window

Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully 

[Suggest Feedback](#)

Test Cases Passed

1120 / 1120

Attempts : Correct / Total

4 / 4

Accuracy : 100%

Time Taken

0.16

Name: - Tirth Kothadiya
Enrollment No. : - 92410133008
Doman: - Microsoft

Easy: - Is Binary Number Multiple of 3

You are given a binary number as a string of characters ('0' and '1'). Your task is to determine whether this binary number is divisible by 3.

Code: -

// User function template for C++

```
class Solution {
```

```
public:
```

```
    bool isDivisible(string& s) {
```

```
        int remainder = 0;
```

```
        for (char c : s) {
```

```
            int bit = c - '0'; // Convert '0'/'1' character to integer 0/1
```

```
            remainder = (remainder * 2 + bit) % 3; // Update the remainder using modulo 3 logic
```

```
        }
```

```
        return remainder == 0; // If final remainder is 0, number is divisible by 3
```

```
    }
```

```
};
```

Output: -

Output Window



Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓

[Suggest Feedback](#)

Test Cases Passed

1111 / 1111

Attempts : Correct / Total

3 / 3

Accuracy : 100%

Time Taken

0.02

Name: - Tirth Kothadiya
Enrollment No. : - 92410133008
Doman: - Microsoft

Easy: - Check for BST

Given the root of a binary tree. Check whether it is a BST or not.

Note: We are considering that BSTs can not contain duplicate Nodes.

A BST is defined as follows:

Code: -

```
class Solution {
public:
    // Function to check whether a Binary Tree is BST or not.
    bool isBST(Node* root) {
        int data;
        Node* left;
        Node* right;

        Node(int val) {
            data = val;
            left = right = nullptr;
        }

    }
};
```

Output:


Name: - Tirth Kothadiya
Enrollment No. : - 92410133008
Doman: - Microsoft

Output Window

Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully 

[Suggest Feedback](#)


Test Cases Passed

1111 / 1111


Attempts : Correct / Total

1 / 8

Accuracy : 12%

Points Scored 

2 / 2

Your Total Score: 8 

Time Taken

0.06

Easy: - Remove every kth node

Given a singly linked list, your task is to remove every kth node from the linked list.

Code: -

// Structure for a singly linked list node

```
struct Node {  
    int data;  
    Node* next;  
    Node(int x) : data(x), next(nullptr) {}  
};
```

```
class Solution {  
public:  
    Node* deleteK(Node* head, int k) {  
        if (k == 1) {  
            // If k == 1, delete all nodes  
            while (head) {  
                Node* temp = head;  
                head = head->next;  
                delete temp;  
            }  
        }  
    }  
};
```


Name: - Tirth Kothadiya
Enrollment No. : - 92410133008
Doman: - Microsoft

```
        return nullptr;
    }

    Node* curr = head;
    Node* prev = nullptr;
    int count = 1;

    while (curr != nullptr) {
        if (count % k == 0) {
            // Delete current node
            prev->next = curr->next;
            delete curr;
            curr = prev->next;
        } else {
            prev = curr;
            curr = curr->next;
        }
        count++;
    }
    return head;
}
```

Output: -

Output Window 

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully 

[Suggest Feedback](#)

Test Cases Passed

1115 / 1115

Attempts : Correct / Total

1 / 2

Accuracy : 50%

Points Scored 

2 / 2

Your Total Score: 10 

Time Taken

0.63

Name: - Tirth Kothadiya
Enrollment No. : - 92410133008
Doman: - Microsoft

Easy: - Reverse a linked list


Given the head of a linked list, the task is to reverse this list and return the reversed head.

Code: -

```
class Solution {
public:
    Node* reverseList(Node* head) {
        Node* prev = nullptr;
        Node* current = head;
        Node* next = nullptr;

        while (current != nullptr) {
            next = current->next; // store next
            current->next = prev; // reverse pointer
            prev = current;      // move prev one step
            current = next;      // move current one step
        }
        return prev; // new head
    }
};
```

Output: -

Output Window 

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully 

[Suggest Feedback](#)

Test Cases Passed

1115 / 1115

Attempts : Correct / Total

1 / 2

Accuracy : 50%

Points Scored 

2 / 2

Your Total Score: 12 

Time Taken

0.16