**Assignment - 3**

**Q1 Given two integer arrays nums1 and nums2, return *an array of their intersection*. Each element in the result must appear as many times as it shows in both arrays and you may return the result in any order.**

**Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]**

**Output: [4,9]**

**Explanation: [9,4] is also accepted.**

Sol:- Code

#include <bits/stdc++.h>

using namespace std;

vector<int> intersect(vector<int>& nums1, vector<int>& nums2) {

    if(nums1.size()>nums2.size()){

        swap(nums1,nums2);

    }

    unordered\_map< int , int >  m;

    for(auto val:nums1){

        m[val]++;

    }

    int k=0;

    for(auto val:nums2){

        if(m[val]>0){

            nums1[k++]=val;

            --m[val];

        }

    }

    return vector<int>(nums1.begin(),nums1.begin()+k);

}

int main()

{

    vector<int> nums1={4,9,5};

    vector<int> nums2={9,4,9,8,4};

    vector<int> ans=intersect(nums1,nums2);

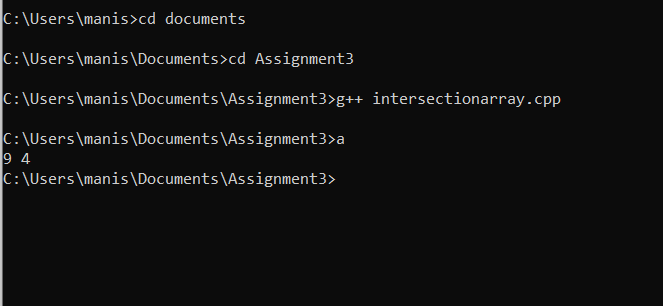
    for(int x:ans)

        cout<<x<<" ";

   return 0;

}

**Output**



**Q.2**Given pointer to the head node of a linked list, the task is to reverse the linked list. We need to reverse the list by changing the links between nodes.

Input: Head of following linked list

1->2->3->4->NULL

Output: Linked list should be changed to,

4->3->2->1->NULL

Sol:- **Code:**

#include <bits/stdc++.h>

using namespace std;

struct Node

{

    int data;

    struct Node\* next;

    Node (int data)

    {

        this->data = data;

        next = NULL;

    }

};

struct LinkedList

{

    Node \*head;

    LinkedList()

    {

        head = NULL;

    }

    void reverse()

    {

        Node \*current = head;

        Node \*prev = NULL, \*next = NULL;

        while (current != NULL)

        {

            next = current->next;

            current->next = prev;

            prev = current;

            current = next;

        }

        head = prev;

    }

    void print()

    {

        struct Node \*temp = head;

        while (temp != NULL)

        {

            cout << temp->data << " ";

            temp = temp->next;

        }

    }

    void push(int data)

    {

        Node \*temp = new Node(data);

        temp->next = head;

        head = temp;

    }

};

int main()

{

    LinkedList ll;

    ll.push(4);

    ll.push(3);

    ll.push(2);

    ll.push(1);

    cout << "Given linked list\n";

    ll.print();

    ll.reverse();

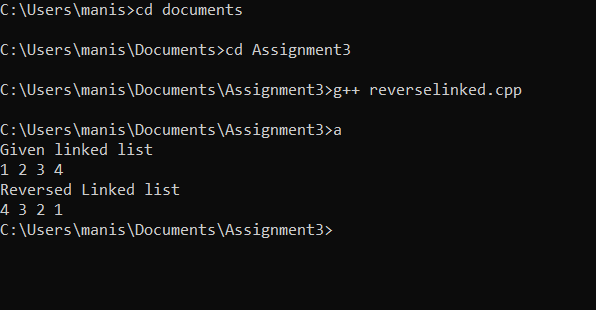
    cout << "\nReversed Linked list\n";

    ll.print();

    return 0;

}

**Output:**



**MCQs**

1. Which stream class is to only write on files?

Sol:- ofstream

2. Which stream class is to only read from files?

Sol:- ifstream

3. Which stream class is used to both read and write on files?

Sol:- fstream

4. Which among following is used to open a file in binary mode?

Sol:- ios::binary

5. ios::trunc is used for?

Sol:- If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one.

6. Which is correct syntax?

Sol:- myfile.open ("example.bin", ios::out);

7. Which among following is correct syntax of closing a file in c++?

Sol:- myfile.close();