Assignment - 3

MCQs :-

Ans→(1) (i) ofstream

Ans→2) (ii) ifstream

Ans→3) (iii) fstream

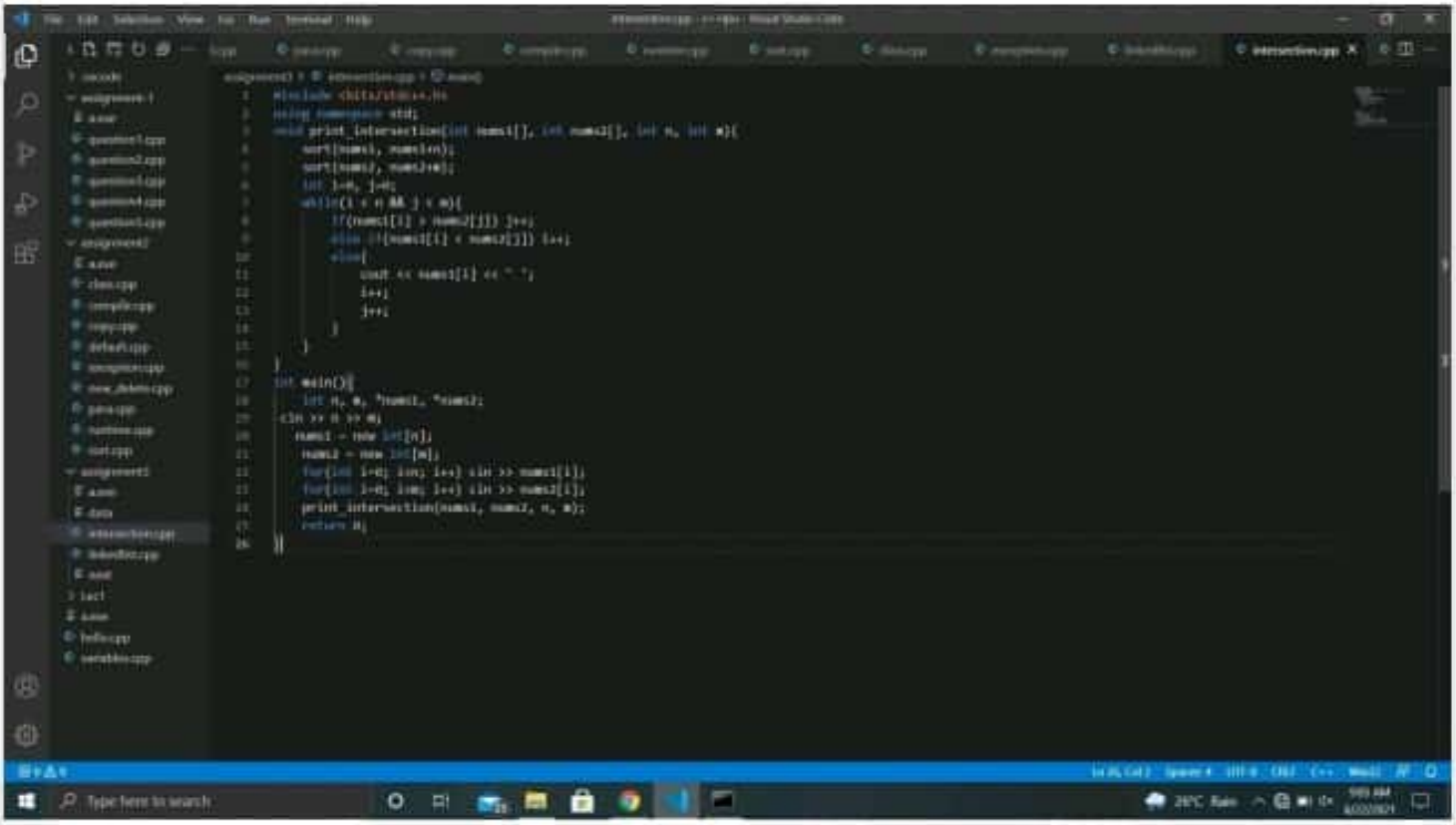
Ans→4) (iv) ios::binary

Ans→5) (ii) If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one.

Ans→6) (ii) myfile.open("example.bin", ios::out);

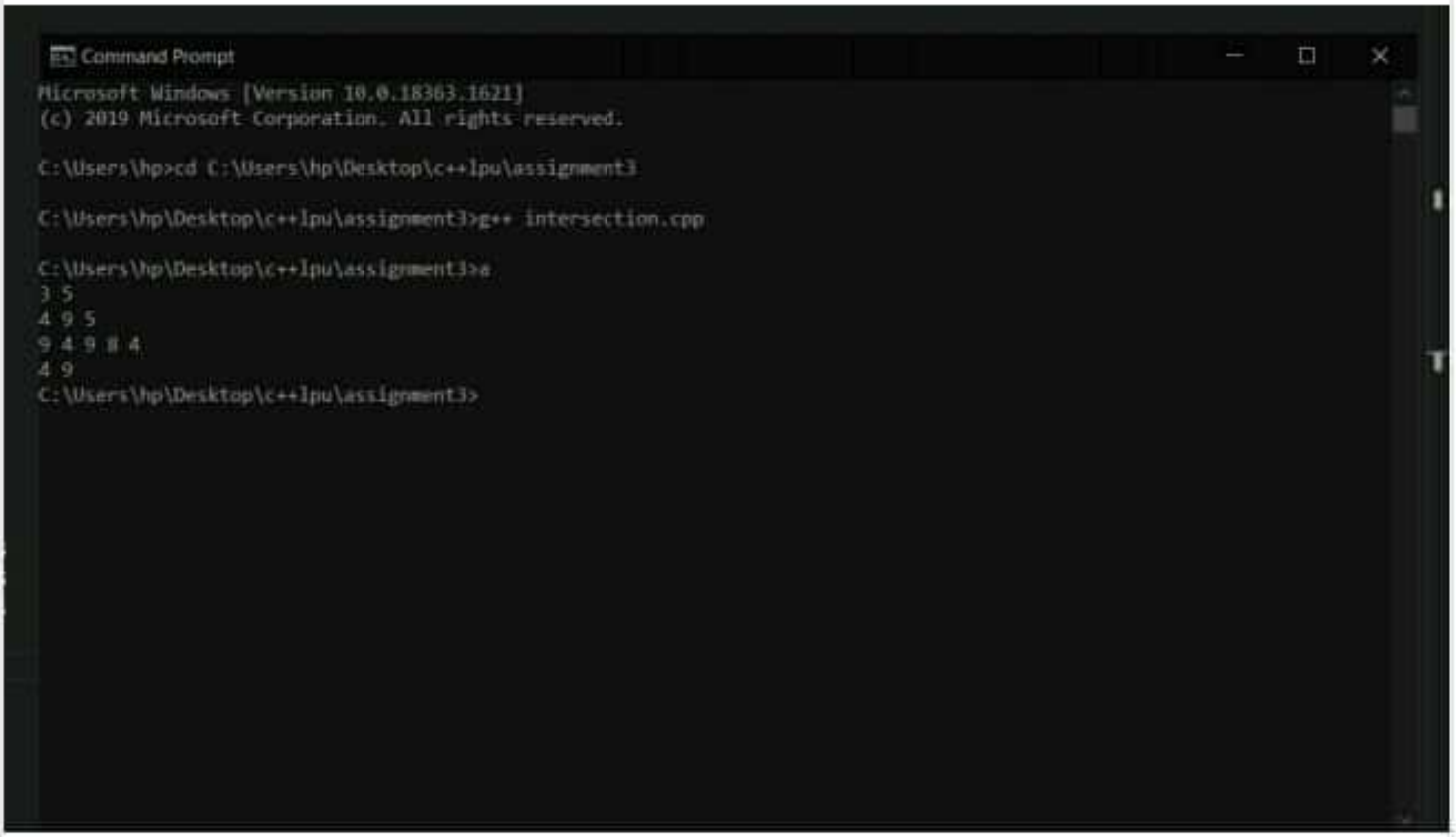
Ans→7) (iv) myfile.close();

Q1) Interesction of array:-



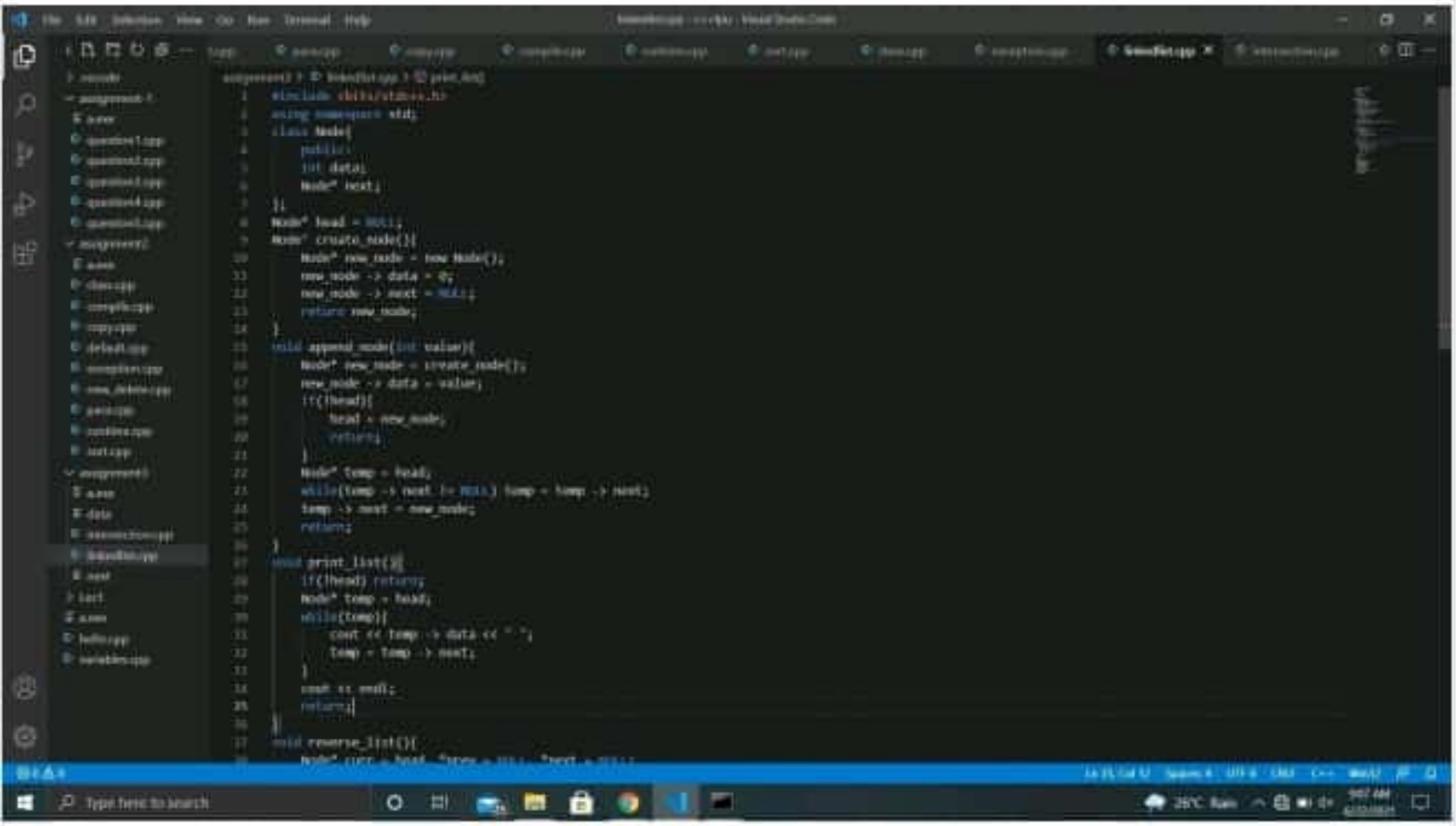
```cpp
#include <bits/stdc++.h>
using namespace std;
void print_intersection(int nums1[], int nums2[], int n, int m){
    sort(nums1, nums1+n);
    sort(nums2, nums2+m);
    int i=0, j=0;
    while(i < n && j < m){
        if(nums1[i] > nums2[j]) j++;
        else if(nums1[i] < nums2[j]) i++;
        else{
            cout << nums1[i] << " ";
            i++;
            j++;
        }
    }
}
int main(){
    int n, m, *nums1, *nums2;
cin >> n >> m;
  nums1 = new int[n];
  nums2 = new int[m];
  for(int i=0; i<n; i++) cin >> nums1[i];
  for(int i=0; i<m; i++) cin >> nums2[i];
  print_intersection(nums1, nums2, n, m);
  return 0;
}
```

Output:-

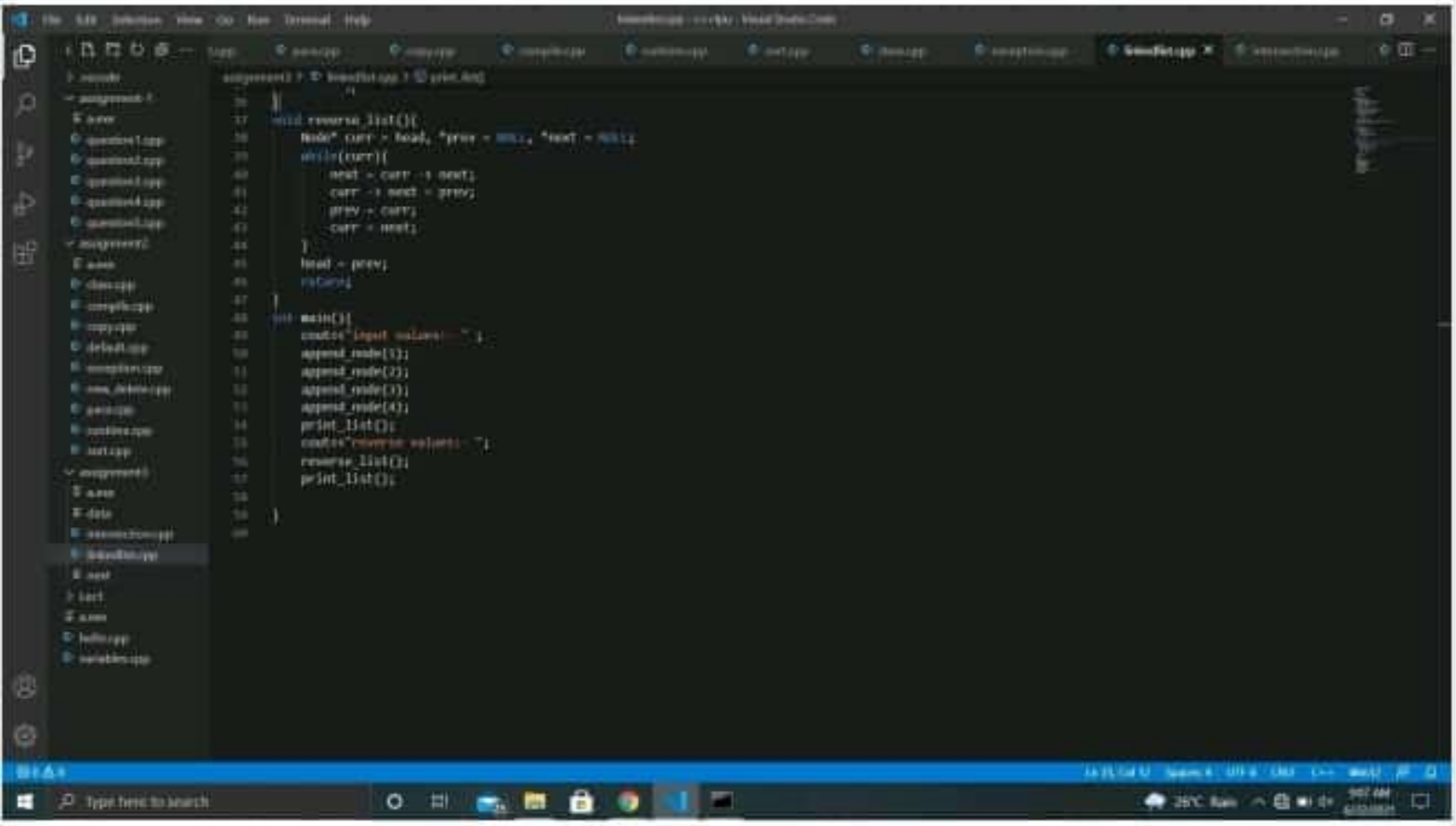


```
Command Prompt
Microsoft Windows [Version 10.0.18363.1621]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\c++lpu\assignment3

C:\Users\hp\Desktop\c++lpu\assignment3>g++ intersection.cpp

C:\Users\hp\Desktop\c++lpu\assignment3>a
3 5
4 9 5
9 4 9 8 4
4 9
C:\Users\hp\Desktop\c++lpu\assignment3>
```

Q2) To reverse the linkedlist:-



```cpp
#include <bits/stdc++.h>
using namespace std;
class Node{
    public:
    int data;
    Node* next;
};
Node* head = NULL;
Node* create_node(){
    Node* new_node = new Node();
    new_node -> data = 0;
    new_node -> next = NULL;
    return new_node;
}
void append_node(int value){
    Node* new_node = create_node();
    new_node -> data = value;
    if(!head){
        head = new_node;
        return;
    }
    Node* temp = head;
    while(temp -> next != NULL) temp = temp -> next;
    temp -> next = new_node;
    return;
}
void print_list(){
    if(!head) return;
    Node* temp = head;
    while(temp){
        cout << temp -> data << " ";
        temp = temp -> next;
    }
    cout << endl;
    return;
}
void reverse_list(){
    Node* curr = head, *prev = NULL, *next = NULL;
    while(curr){
        next = curr -> next;
        curr -> next = prev;
        prev = curr;
        curr = next;
    }
    head = prev;
    return;
}
int main(){
    cout<<"input values: " ;
    append_node(1);
    append_node(2);
    append_node(3);
    append_node(4);
    print_list();
    cout<<"reverse values: ";
    reverse_list();
    print_list();

}
```



Output:-