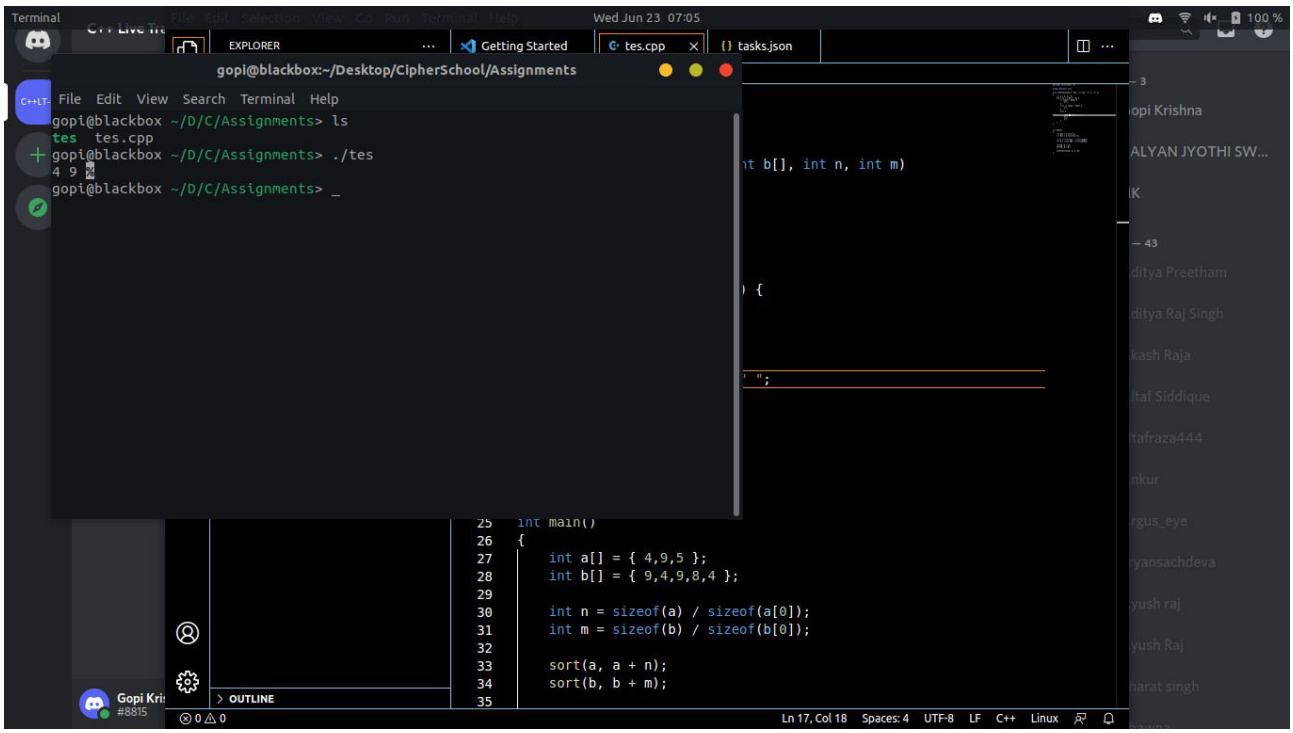
Assignment-3

Q1.
Answer :

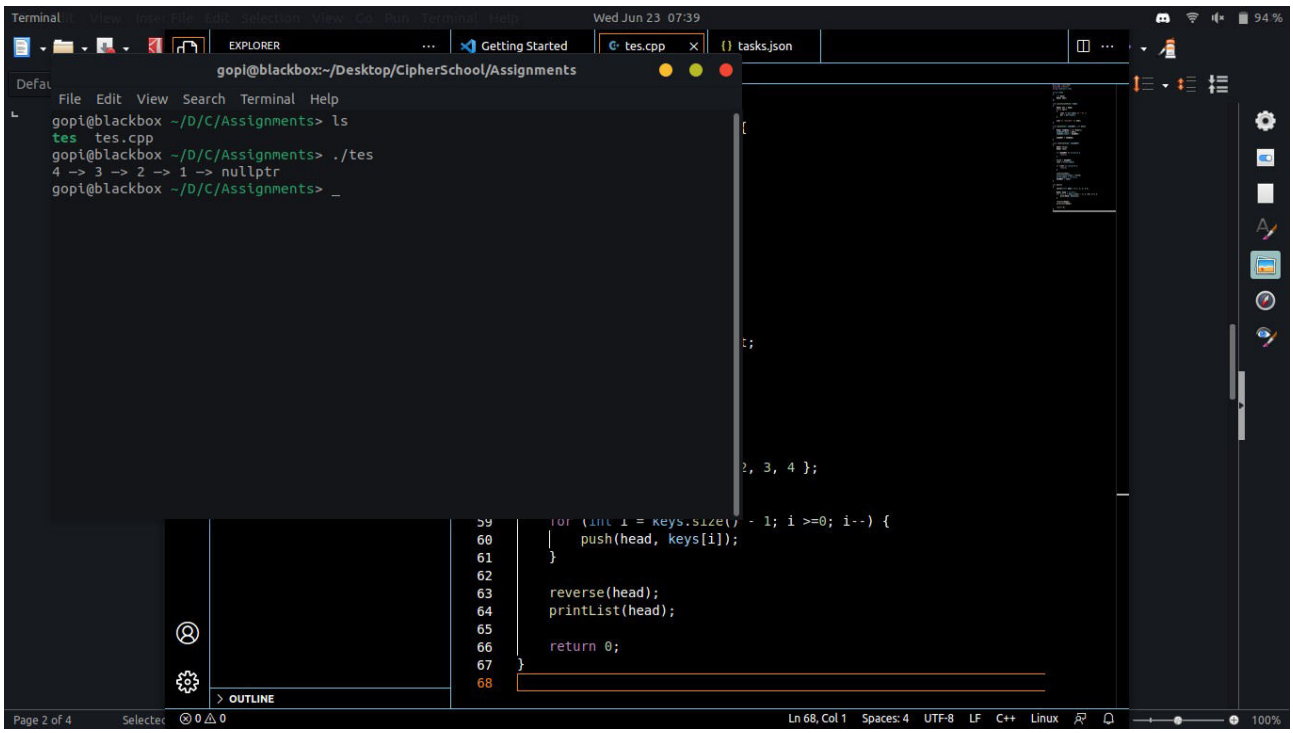
```
#include <bits/stdc++.h>

using namespace std;

void intersection(int a[], int b[], int n, int m)
{
	int i = 0, j = 0;
	while (i < n && j < m) {
		if (a[i] > b[j]) {
			j++;
		}
		else if (b[j] > a[i]) {
			i++;
		}
		else {

			cout << a[i] << " ";
			i++;
			j++;
		}
	}
}


int main()
{
	int a[] = { 4,9,5 };
	int b[] = { 9,4,9,8,4 };

	int n = sizeof(a) / sizeof(a[0]);
	int m = sizeof(b) / sizeof(b[0]);

	sort(a, a + n);
	sort(b, b + m);

	intersection(a, b, n, m);
}
```



Q2.
Answer :

```
#include <iostream>
#include <vector>
using namespace std;

struct Node
{
    int data;
    Node* next;
};

void printList(Node* head)
{
    Node* ptr = head;
    while (ptr)
    {
        cout << ptr->data << " —> ";
        ptr = ptr->next;
    }
```

```
    cout << "nullptr" << endl;
}

void push(Node* &headRef, int data)
{
    Node* newNode = new Node();
    newNode->data = data;
    newNode->next = headRef;

    headRef = newNode;
}

void reverse(Node* &headRef)
{
    Node* first;
    Node* rest;

    if (headRef == nullptr) {
        return;
    }

    first = headRef;
    rest = first->next;

    if (rest == nullptr) {
        return;
    }

    reverse(rest);
    first->next->next = first;
    first->next = nullptr;
    headRef = rest;
}

int main()
{
    vector<int> keys = { 1, 2, 3, 4 };

    Node* head = nullptr;
    for (int i = keys.size() - 1; i >=0; i--) {
        push(head, keys[i]);
    }

    reverse(head);
    printList(head);

    return 0;
}
```



MCQ ;

1) ofstream

2) ifstream

3) fstream

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

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

6) myfile.close();