Assignment - 3

Roshan kumar

Q1. Answer

```
#include <bits/stdc++.h>
using namespace std
void intersection (int a[], int b[], int n, int
{
  int i=0, j=0;
  while (i<n && j<m1)
      {
        if (a[i] > b[j])
            {
              j++;
            }
      else if (b[j] > a[i])
          {
            i++;
          }
      else
          {
          cout << a[i] << " ";
          i++;
          j++
          }
      }
}
```

P.T.O.

```
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)
```

P.T.O

```
{
    Node* ptr = head;
    while (ptr)
{
    cout << ptr->data<<" ->";
    ptr = ptr-> next;
}
    cout <<"nullptr" <<endl;
}
void push (Node*& headRef ; int data )
{
    Node* newNode = newNode ();
    newNode ->data = data;
    newNode -> Node* = headRef
    headRef = new Node;
}
void reverse (Node* &headRef)
{
    Node* first;
    Node* rest;
    if
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 = key.size()-1; i>=0; i--)
          {
            push(head, keys[i]);
          }
      reverse (head)
      printList (head)
      return 0;

    }
```