Assignment 2

Name : Om D. Tambat

Enrollment No. : BT22CSH042

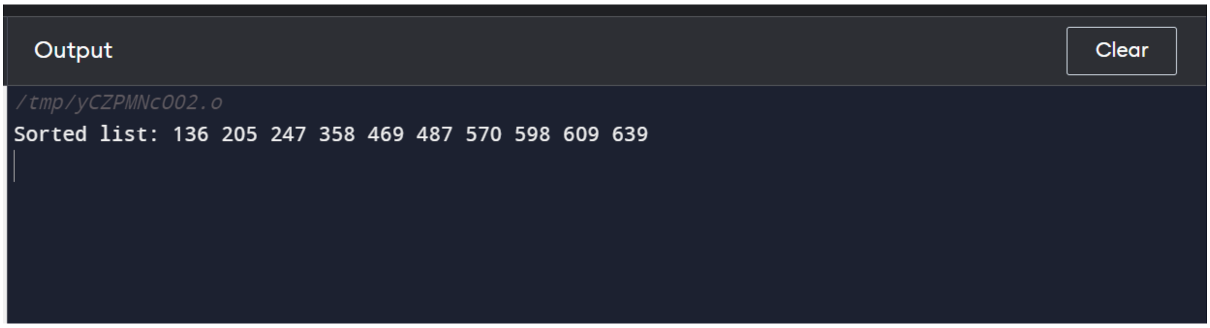
Q1.

Code :

```
#include <iostream>
#include <vector>
#include <algorithm>  // Add this line for the max_element function
using namespace std;

void countingSort(vector<int>& arr, int exp) {
   int n = arr.size();
   vector<int> output(n);
   vector<int> count(10, 0);
   for (int num : arr) {
      int index = (num / exp) % 10;
      count[index]++;
   }
   for (int i = 1; i < 10; ++i) {
      count[i] += count[i - 1];
   }
   for (int i = n - 1; i >= 0; --i) {
      int num = arr[i];
      int index = (num / exp) % 10;
      output[count[index] - 1] = num;
      count[index]--;
```

```
    }
    for (int i = 0; i < n; ++i) {
        arr[i] = output[i];
    }
}

void radixSort(vector<int>& arr) {
    int max_num = *max_element(arr.begin(), arr.end());
    int exp = 1;
    while (max_num / exp > 0) {
        countingSort(arr, exp);
        exp *= 10;
    }
}

int main() {
    vector<int> input_list = {136, 487, 358, 469, 570, 247, 598, 639, 205, 609};
    radixSort(input_list);
    cout << "Sorted list: ";
    for (int num : input_list) {
        cout << num << " ";
    }
    cout << endl;
    return 0;
}
```

Output :



```
/tmp/yCZPMNcO02.o
Sorted list: 136 205 247 358 469 487 570 598 609 639
```

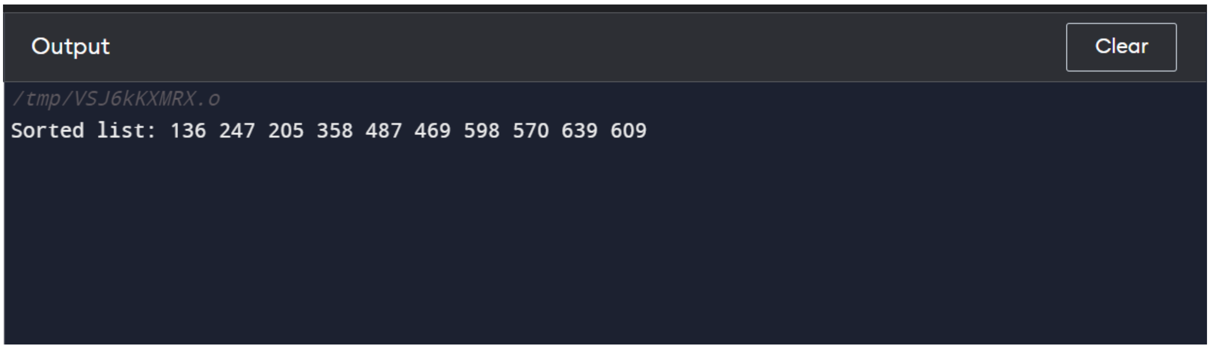
Q2.

Code :

```
#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

class Node {

public:

 int data;

 Node* next;

 Node(int value) : data(value), next(nullptr) {}

};

void countingSortLinkedList(Node* &head, int exp) {

 if (!head) return;

 vector<Node*> buckets(10, nullptr);

 Node* current = head;

 while (current != nullptr) {

 int index = (current->data / exp) % 10;

 Node* next = current->next;

 current->next = buckets[index];
 buckets[index] = current;
 current = next;
 }
 head = nullptr;
 Node* tail = nullptr;
 for (int i = 0; i < 10; ++i) {
 if (buckets[i] != nullptr) {
 if (head == nullptr) {
 head = buckets[i];
 tail = buckets[i];
 } else {
 tail->next = buckets[i];
 tail = tail->next;
 }
 Node* temp = buckets[i];
 while (temp->next != nullptr) {
 temp = temp->next;
 }
 tail = temp;
 }
 }
}
void radixSortLinkedList(Node* &head) {
 if (!head) return;
 int max_num = head->data;
 Node* current = head->next;
 while (current != nullptr) {
 max_num = max(max_num, current->data);
 current = current->next;
 }
 int exp = 1;
 while (max_num / exp > 0) {
 countingSortLinkedList(head, exp);
 exp *= 10;
 }
}
void printList(Node* head) {
 while (head != nullptr) {
 cout << head->data << " ";
 head = head->next;
 }
 cout << endl;
}
int main() {
 Node* head = nullptr;
 Node* tail = nullptr;
 int elements[] = {136, 487, 358, 469, 570, 247, 598, 639, 205, 609};
 int n = sizeof(elements) / sizeof(elements[0]);
 for (int i = 0; i < n; ++i) {
 int num = elements[i];
 Node* newNode = new Node(num);
 if (!head) {
```

```
 head = newNode;
 tail = newNode;
 } else {
 tail->next = newNode;
 tail = newNode;
 }
 }
 radixSortLinkedList(head);
 cout << "Sorted list: ";
 printList(head);
 // Free memory
 while (head != nullptr) {
 Node* temp = head;
 head = head->next;
 delete temp;
return 0;
}
}
```

Output :



```
/tmp/VSJ6kKXMRX.o
Sorted list: 136 247 205 358 487 469 598 570 639 609
```