Question – 1 :

```
#include <bits/stdc++.h>
using namespace std;

int Vaccine = 0;

class Region {
public:
    int Region_Id;
    string Region_Name;
    long population;
    int Riskfactor;
    bool need_vaccine;
    long vaccinated;

    Region(int Region_Id, int Region_Name, int population, int Riskfactor) {
        this->Region_Id = Region_Id;
        this->Region_Name = Region_Name;
        this->population = population;
        this->Riskfactor = Riskfactor;
        this->need_vaccine = true;
        vaccinated = 0;
    }
};

void Add_Region_Data(int Region_Id, int Region_Name, int population, int Riskfactor) {
    Region *region = new Region(Region_Id, Region_Name, population, Riskfactor);
    cout << "Region added successfully!" << endl;
    return;
}
```

```
void Update_Vaccine(int change) {
   cout << change << "doeses added into the vaccine" << endl;
   Vaccine += change;
   return;
}

void Greedy_Allocation_of_Vaccines(vector<Region>& Regions, vector <pair<string, int>> &report) {
   sort(Regions.begin(), Regions.end(), [](const Region& a, const Region& b) {

      if (a.Riskfactor == b.Riskfactor) {
         return a.population > b.population;
      }

      return a.Riskfactor > b.Riskfactor;
      });

   for (auto& region : Regions) {
      if (Vaccine > 0 && region.need_vaccine) {

         if (Vaccine >= region.population - region.vaccinated) {
            region.need_vaccine = false;
            cout << "Region " << region.Region_Name << " received " << region.population -
region.vaccinated << " doses of vaccine." << endl;
            report.push_back({ region.Region_Name  , region.population - region.vaccinated });
            region.vaccinated = region.population;
            Vaccine -= region.population;
         }

         else {
            cout << "Region " << region.Region_Name << " received " << Vaccine << " doses of vaccine."
<< endl;
```

```
            report.push_back({ region.Region_Name  , Vaccine });
            region.vaccinated += Vaccine;
            Vaccine = 0;
        }
    }
  }

  return;
}

void Generate_Distribution_Report(vector<Region>& Regions, vector <pair<string, int>>& report,
vector<string, int>& needed) {

  for (auto& region : Regions) {
    if (region.need_vaccine) {
      needed.push_back({ region.Region_Name, region.population });
    }
  }

  for (auto& i : report) {
    cout << "Region -  "<< i.first << " Received " << i.second << " Vaccines." << endl;
  }

  for (auto& region : Regions) {
    if (region.need_vaccine) {
      needed.push_back({ region.Region_Name  , region.population - region.vaccinated });

      cout << "The Region" << region.Region_Name << " needed " << region.population -
region.vaccinated << " Vaccines ." << endl;
    }
  }
```

```
    cout << "The Remaining vaccines are " << Vaccine << endl;


    for (auto& region : Regions) {
        cout << "Region -  " << region.Region_Name << " has " << region.vaccinated << " and " <<
(((region.population - region.vaccinated) / region.population) * 100) << endl;
    }
}


int main() {
    vector<Region> Regions;
    vector <pair<string, int>> report;
    vector<string, int> needed;


    return 0;
}
```

Question – 2 :

```
#include <bits/stdc++.h>
using namespace std;


void Merge(vector <pair<int, string>>& Marks, int low, int mid, int high) {


    int lower_bound = mid - low + 1;
    int upper_bound = high - mid;


    vector<pair<int, string>> Left_Array(lower_bound);
    vector<pair<int, string>> Right_Array(upper_bound);


    for (int i = 0; i < lower_bound; i++) {
```

```
    Left_Array[i] = Marks[low + i];
}

for (int i = 0; i < upper_bound; i++) {
    Right_Array[i] = Marks[mid + 1 + i];
}

int i = 0;
int j = 0;
int k = low;

while (i < lower_bound && j < upper_bound) {
    if (Left_Array[i] <= Right_Array[j]) {
        Marks[k] = Left_Array[i];
        i++;
    }

    else {
        Marks[k] = Right_Array[j];
        j++;
    }

    k++;
}

while (i < lower_bound) {
    Marks[k] = Left_Array[i];
    i++;
    k++;
}
```

```
    while (j < upper_bound) {
        Marks[k] = Right_Array[j];
        j++;
        k++;
    }
}

void Merge_Sort(vector <pair<int, string>>& Marks, int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        Merge_Sort(Marks, left, mid);
        Merge_Sort(Marks, mid + 1, right);
        Merge(Marks, left, mid, right);
    }
}

// pair<int, string> Max_Marks(vector <pair<int, string>>& Marks) {
//     pair<int, string> Max_Student = {INT_MIN , ""};

//     for (auto& mark : Marks) {
//         if (mark.first > Max_Student.first) {
//             Max_Student = mark;
//         }
//     }
//     return Max_Student;
// }

// pair<int, string> Min_Marks(vector <pair<int, string>>& Marks) {
//     pair<int, string> Min_Student = { INT_MAX , "" };

//     for (auto& mark : Marks) {
```

```
//        if (mark.first < Min_Student.first) {
//            Min_Student = mark;
//        }
//    }
//    return Min_Student;
// }

int main() {

    vector<pair<int, string>> Marks = { {95, "Alice"}, {85, "Bob"}, {90, "Charlie"}, {88, "David"} };

    int left = 0;
    int right = Marks.size() - 1;
    Merge_Sort(Marks, left, right);

    // pair<int, string> max_student = Marks.back();
    // pair<int, string> min_student = Marks.front();

    // cout << "Student with highest marks: " << max_student.second << " with " << max_student.first
<< " marks" << endl;
    // cout << "Student with lowest marks: " << min_student.second << " with " << min_student.first <<
" marks" << endl;

    int min_marks = Marks[0].first;
    vector<string> min_studens;

    for (int i = 0; i < Marks.size(); i++) {
        if (Marks[i].first == min_marks) {
            min_studens.push_back(Marks[i].second);
        }
    }
```

```
    int max_marks = Marks.back().first;
    vector<string> max_studens;

    for (int i = Marks.size(); i >= 0 ; i--) {
        if (Marks[i].first == max_marks) {
            max_studens.push_back(Marks[i].second);
        }
    }

    cout << "The Maximum Marks are " << max_marks << endl;
    cout << "The Student Sccored Max_Marks are :- ";

    for (string name : max_studens) {
        cout << name << " , ";
    }

    cout << endl;

    cout << "The Minimum Marks are " << min_marks << endl;
    cout << "The Student Sccored Max_Marks are :- ";

    for (string name : min_studens) {
        cout << name << " , ";
    }

    cout << endl;

    return 0;
}
```

Outpput :-

```
                                                                                > cd "d:\Aryan Data\Useful
Data\Semester - 5\Design-and-Analysis-of-Algorithms\Mid - Sem Practicle Exam 17-10-2024\" ; if ($?) { g++ 92200133030_Question2.cpp -o 922
133030_Question2 } ; if ($?) { .\92200133030_Question2 }
The Maximum Marks are 95
The Student Sccored Max_Marks are :- Alice ,
The Minimum Marks are 85
The Student Sccored Max_Marks are :- Bob ,
PS D:\Aryan Data\Usefull Data\Semester - 5\Design-and-Analysis-of-Algorithms\Mid - Sem Practicle Exam 17-10-2024>
```

```
PS D:\Aryan Data\Usefull Data\Semester - 5\Design-and-Analysis-of-Algorithms\Mid - Sem Practicle Exam 17-10-2024> cd "d:\Aryan Data\Usefull
Data\Semester - 5\Design-and-Analysis-of-Algorithms\Mid - Sem Practicle Exam 17-10-2024\" ; if ($?) { g++ 92200133030_Question2.cpp -o 92200
133030_Question2 } ; if ($?) { .\92200133030_Question2 }
The Maximum Marks are 95
The Student Sccored Max_Marks are :- Alice ,
The Minimum Marks are 85
The Student Sccored Max_Marks are :- Bob , Charlie ,
PS D:\Aryan Data\Usefull Data\Semester - 5\Design-and-Analysis-of-Algorithms\Mid - Sem Practicle Exam 17-10-2024>
```