**OOP Lab No: 01**

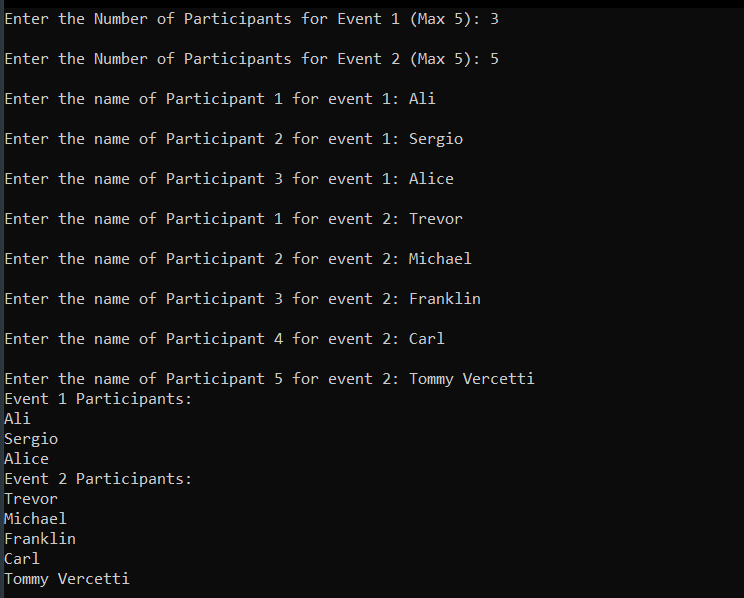
**Name: Ali Rooman**

**Roll No: 24K-0792**

**In Lab Tasks**

Q1:

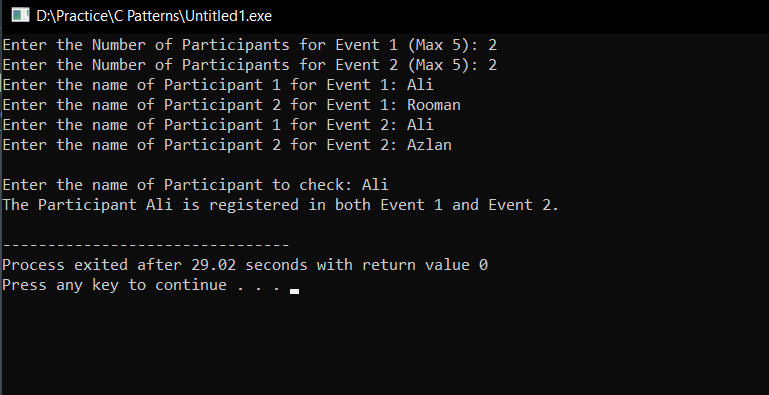
|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  int main()  {      int Participant1, Participant2;      cout<<"Enter the Number of Participants for Event 1 (Max 5): ";      cin>>Participant1;      cout<<"\nEnter the Number of Participants for Event 2 (Max 5): ";      cin>>Participant2;        cin.ignore();      string Event1[Participant1], Event2[Participant2];      for (int i=0;i<Participant1;i++)      {          cout<<"\nEnter the name of Participant "<<i+1<<" for event 1: ";          getline(cin,Event1[i]);      }      for (int i=0;i<Participant2;i++)      {          cout<<"\nEnter the name of Participant "<<i+1<<" for event 2: ";          getline(cin,Event2[i]);      }      cout<<"Event 1 Participants: \n";      for (int i=0;i<Participant1;i++)      {          cout<<Event1[i]<<endl;      }      cout<<"Event 2 Participants: \n";      for (int i=0;i<Participant2;i++)      {          cout<<Event2[i]<<endl;      }      return 0;  } |

Output

Q2

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  void checkParticipant(string Event1[], string Event2[], int Participant1, int Participant2) {  string SearchName;  cout<<"\nEnter the name of Participant to check: ";  getline(cin, SearchName);  int found1=0, found2=0;  for(int i=0; i<Participant1; i++) {  if(SearchName == Event1[i]) {  found1=1;  break;  }  }  for(int i=0; i<Participant2; i++) {  if(SearchName == Event2[i]) {  found2=1;  break;  }  }  if(found1==1 && found2==1)  cout<<"The Participant "<<SearchName<<" is registered in both Event 1 and Event 2.\n";  else if(found1==1)  cout<<"The Participant "<<SearchName<<" is registered in Event 1.\n";  else if(found2==1)  cout<<"The Participant "<<SearchName<<" is registered in Event 2.\n";  else  cout<<"The Participant "<<SearchName<<" is not registered in either event.\n";  }  int main() {  int Participant1, Participant2;  cout<<"Enter the Number of Participants for Event 1 (Max 5): ";  cin>>Participant1;  cout<<"Enter the Number of Participants for Event 2 (Max 5): ";  cin>>Participant2;  cin.ignore();  string Event1[Participant1], Event2[Participant2];  for(int i=0; i<Participant1; i++) {  cout<<"Enter the name of Participant "<<i+1<<" for Event 1: ";  getline(cin, Event1[i]);  }  for(int i=0; i<Participant2; i++) {  cout<<"Enter the name of Participant "<<i+1<<" for Event 2: ";  getline(cin, Event2[i]);  }  checkParticipant(Event1, Event2, Participant1, Participant2);  return 0;  } |

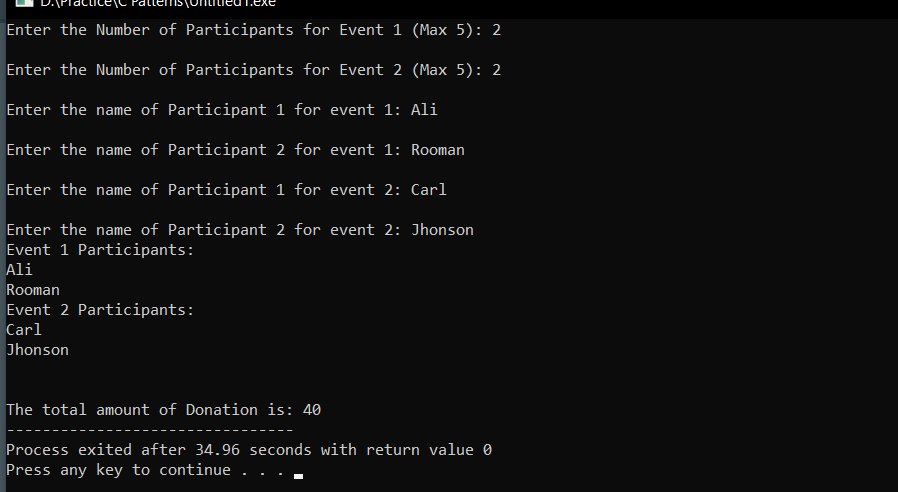
Output



Q3

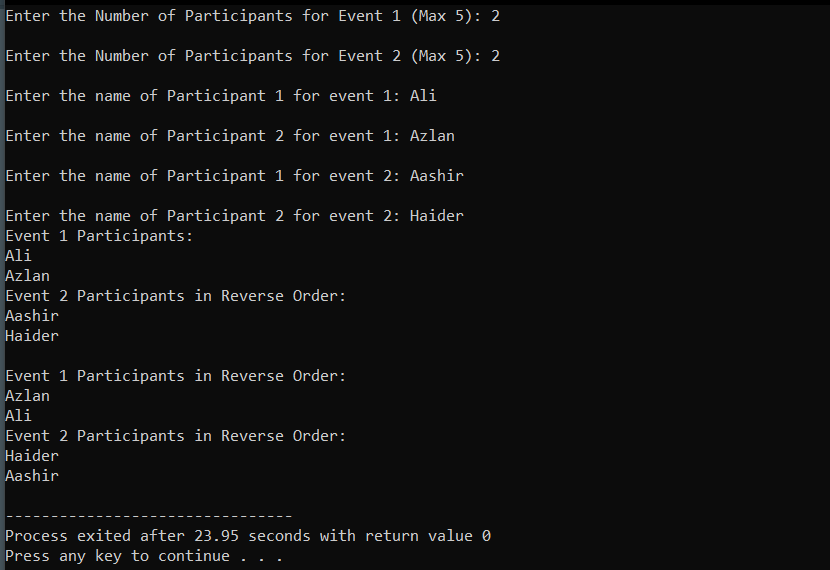
|  |
| --- |
| #include <iostream>  #include <string>  using *namespace* std;  *int* main()  {  *int* Participant1, Participant2;      cout<<"Enter the Number of Participants for Event 1 (Max 5): ";      cin>>Participant1;      cout<<"\nEnter the Number of Participants for Event 2 (Max 5): ";      cin>>Participant2;        cin.ignore();      string Event1[Participant1], Event2[Participant2];      for (*int* i=0;i<Participant1;i++)      {          cout<<"\nEnter the name of Participant "<<i+1<<" for event 1: ";          getline(cin,Event1[i]);      }      for (*int* i=0;i<Participant2;i++)      {          cout<<"\nEnter the name of Participant "<<i+1<<" for event 2: ";          getline(cin,Event2[i]);      }      cout<<"Event 1 Participants: \n";      for (*int* i=0;i<Participant1;i++)      {          cout<<Event1[i]<<endl;      }      cout<<"Event 2 Participants: \n";      for (*int* i=0;i<Participant2;i++)      {          cout<<Event2[i]<<endl;      }    *int* totalParticipants = Participant1 + Participant2;  *int* totalDontation = totalParticipants \* 10;      cout<<"\n\nThe total amount of Donation is: "<<totalDontation;      return 0;  } |

Output



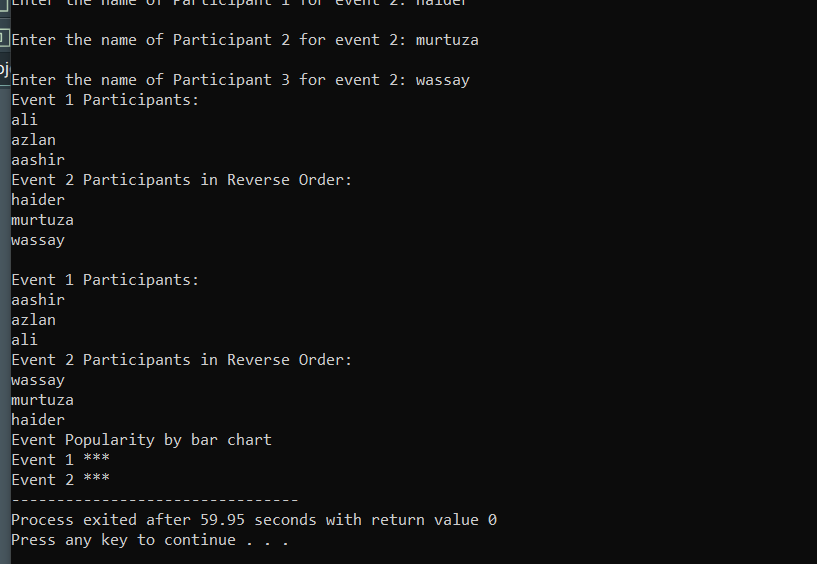
Q4

|  |
| --- |
| #include <iostream>  #include <string>  using *namespace* std;  *void* revOrder(string *Event1*[],string *Event2*[],*int* *Participant1*,*int* *Participant2*){      cout<<"\nEvent 1 Participants in Reverse Order: \n";      for (*int* i=Participant1-1;i>=0;i--)      {          cout<<Event1[i]<<endl;      }      cout<<"Event 2 Participants in Reverse Order: \n";      for (*int* i=Participant2-1;i>=0;i--)      {          cout<<Event2[i]<<endl;      }  }  *int* main()  {  *int* Participant1, Participant2;      cout<<"Enter the Number of Participants for Event 1 (Max 5): ";      cin>>Participant1;      cout<<"\nEnter the Number of Participants for Event 2 (Max 5): ";      cin>>Participant2;        cin.ignore();      string Event1[Participant1], Event2[Participant2];      for (*int* i=0;i<Participant1;i++)      {          cout<<"\nEnter the name of Participant "<<i+1<<" for event 1: ";          getline(cin,Event1[i]);      }      for (*int* i=0;i<Participant2;i++)      {          cout<<"\nEnter the name of Participant "<<i+1<<" for event 2: ";          getline(cin,Event2[i]);      }      revOrder(Event1,Event2,Participant1,Participant2);      return 0;  } |

Output

Q5

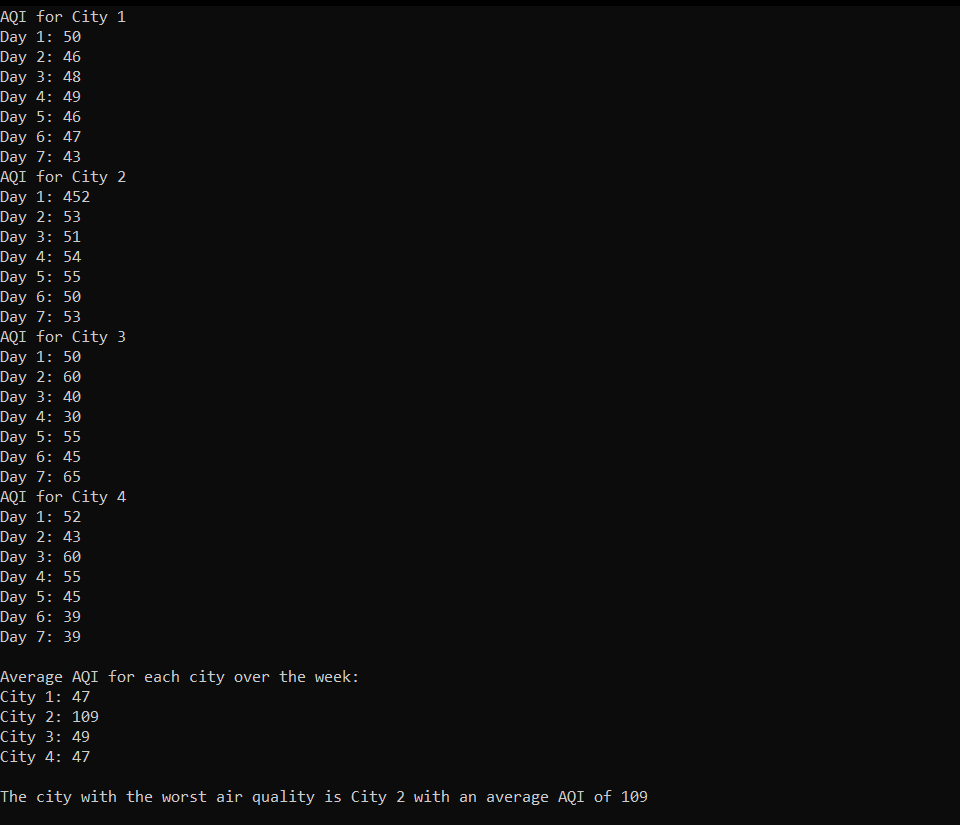
|  |
| --- |
| #include <iostream>  #include <string>  using *namespace* std;  *void* barChart(string *Event1*[],string *Event2*[],*int* *Participant1*,*int* *Participant2*){      cout<<"Event Popularity by bar chart"<<endl;      cout<<"Event 1 ";      for(*int* i=0;i<Participant1;i++){          cout<<"\*";      }      cout<<"\n";      cout<<"Event 2 ";      for(*int* i=0;i<Participant2;i++){          cout<<"\*";      }  }  *int* main(){  *int* Participant1, Participant2;      cout<<"Enter the Number of Participants for Event 1 (Max 5): ";      cin>>Participant1;      cout<<"\nEnter the Number of Participants for Event 2 (Max 5): ";      cin>>Participant2;        cin.ignore();      string Event1[Participant1], Event2[Participant2];      for (*int* i=0;i<Participant1;i++)      {          cout<<"\nEnter the name of Participant "<<i+1<<" for event 1: ";          getline(cin,Event1[i]);      }      for (*int* i=0;i<Participant2;i++)      {          cout<<"\nEnter the name of Participant "<<i+1<<" for event 2: ";          getline(cin,Event2[i]);      }      cout<<"Event 1 Participants: \n";      for (*int* i=0;i<Participant1;i++)      {          cout<<Event1[i]<<endl;      }      cout<<"Event 2 Participants: \n";      for (*int* i=0;i<Participant2;i++)      {          cout<<Event2[i]<<endl;      }        barChart(Event1,Event2,Participant1,Participant2);      return 0;  } |

Output

**Take Home Tasks**

Q1

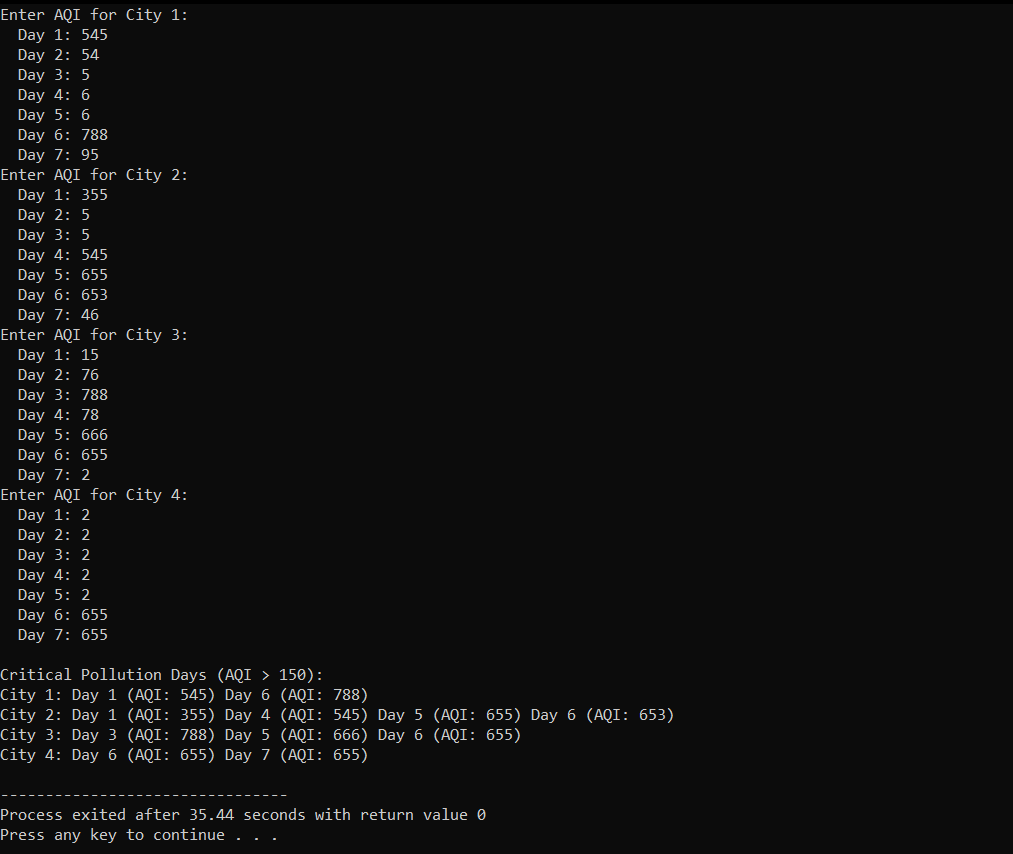
|  |
| --- |
| #include <iostream>  using *namespace* std;  *void* worstAverage(*float* *averages*[],*int* *cities*){  *int* index = 0;  *float* worstAverage = averages[0];      for(*int* i=0;i<cities;i++){          if (averages[i]>worstAverage){              worstAverage = averages[i];              index = i;          }      }      cout << "\nAverage AQI for each city over the week:\n";      for (*int* i = 0; i < cities; i++) {          cout << "City " << i + 1 << ": " << averages[i] << endl;      }      cout << "\nThe city with the worst air quality is City " << index + 1      << " with an average AQI of " << worstAverage << "\n";  }  *int* main() {  *int* cities = 4;  *int* days = 7;  *int* AQI[cities][days];  *float* averages[cities];      for (*int* i=0;i<cities;i++){          cout<<"AQI for City "<<i+1<<endl;          for (*int* j=0;j<days;j++){              cout<<"Day "<<j+1<< ": ";              cin>>AQI[i][j];          }      }      for (*int* i=0;i<cities;i++){  *int* sum = 0;          for (*int* j=0;j<days;j++){              sum = sum + AQI[i][j];          }          averages[i] = sum/days;      }      worstAverage(averages,cities);      return 0;  } |

Output

Q2

|  |
| --- |
| #include <iostream>  using *namespace* std;  *void* criticalDay(*int* *AQI*[4][7], *int* *cities*, *int* *days*) {      cout << "\nCritical Pollution Days (AQI > 150):\n";      for (*int* i = 0; i < cities; i++) {  *bool* dayCritical = false;          cout << "City " << i + 1 << ": ";          for (*int* j = 0; j < days; j++) {              if (AQI[i][j] > 150) {                  cout << "Day " << j + 1 << " (AQI: " << AQI[i][j] << ") ";                  dayCritical = true;              }          }          if (!dayCritical) {              cout << "No critical pollution days.";          }          cout << endl;      }  }  *int* main() {  *const* *int* cities = 4, days = 7;  *int* AQI[cities][days];      for (*int* i = 0; i < cities; i++) {          cout << "Enter AQI for City " << i + 1 << ":\n";          for (*int* j = 0; j < days; j++) {              cout << "  Day " << j + 1 << ": ";              cin >> AQI[i][j];          }      }      criticalDay(AQI, cities, days);      return 0;  } |

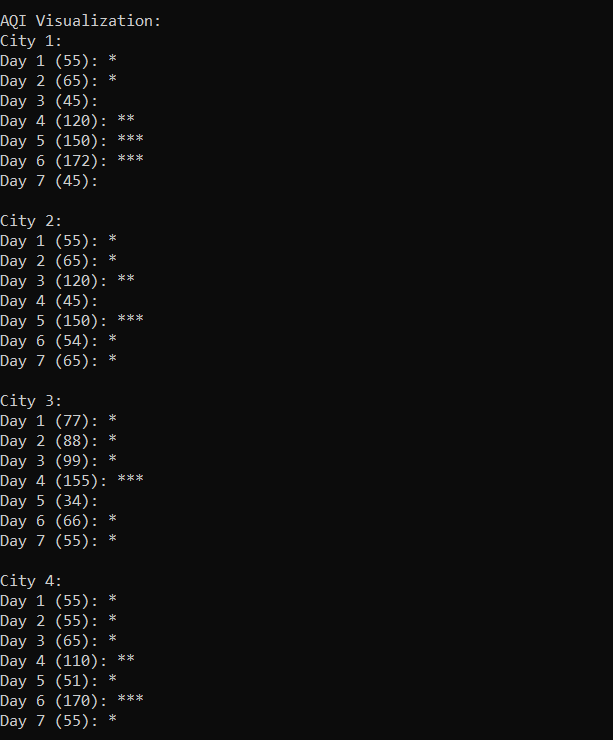
Output



Q3

|  |
| --- |
| #include <iostream>  using *namespace* std;  *void* visualizeAQI(*int* *AQI*[4][7], *int* *cities*, *int* *days*) {      cout<<"\nAQI Visualization:\n";      for(*int* i=0;i<cities;i++){          cout<<"City "<<i+1<<":\n";          for(*int* j=0;j<days;j++){              cout<<"Day "<<j+1<<" ("<<AQI[i][j]<<"): ";  *int* stars=AQI[i][j]/50;              for(*int* k=0;k<stars;k++)cout<<"\*";              cout<<endl;          }          cout<<endl;      }  }  *int* main() {  *const* *int* cities=4,days=7;  *int* AQI[cities][days];      for(*int* i=0;i<cities;i++){          cout<<"Enter AQI for City "<<i+1<<":\n";          for(*int* j=0;j<days;j++){              cout<<"Day "<<j+1<<":";              cin>>AQI[i][j];          }      }      visualizeAQI(AQI,cities,days);      return 0;  } |

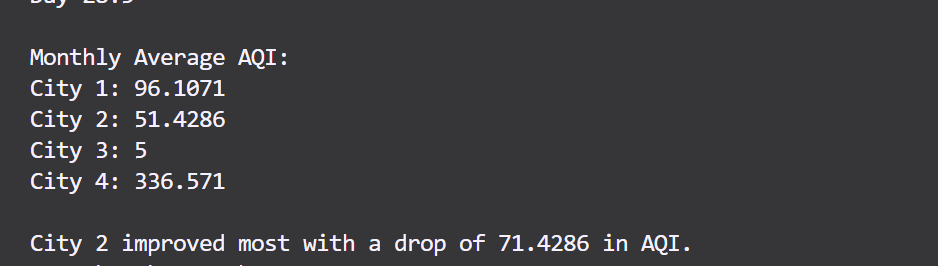
Output



Q4

|  |
| --- |
| #include <iostream>  using *namespace* std;  *void* calcAvg(*int* *AQI*[4][28], *int* *cities*, *int* *days*, *float* *avg*[4]) {      cout<<"\nMonthly Average AQI:\n";      for(*int* i=0;i<cities;i++){  *int* sum=0;          for(*int* j=0;j<days;j++)sum+=AQI[i][j];          avg[i]=sum/(*float*)days;          cout<<"City "<<i+1<<": "<<avg[i]<<endl;      }  }  *void* findBestDrop(*int* *AQI*[4][28], *int* *cities*) {  *float* maxDrop=0;  *int* bestCity=-1;      for(*int* i=0;i<cities;i++){  *float* firstWeek=0,lastWeek=0;          for(*int* j=0;j<7;j++)firstWeek+=AQI[i][j];          for(*int* j=21;j<28;j++)lastWeek+=AQI[i][j];          firstWeek/=7.0,lastWeek/=7.0;  *float* drop=firstWeek-lastWeek;          if(drop>maxDrop){              maxDrop=drop;              bestCity=i+1;          }      }      if(bestCity!=-1)cout<<"\nCity "<<bestCity<<" improved most with a drop of "<<maxDrop<<" in AQI.\n";      else cout<<"\nNo major improvement in air quality.\n";  }  *int* main() {  *const* *int* cities=4,days=28;  *int* AQI[cities][days];  *float* avg[cities];      for(*int* i=0;i<cities;i++){          cout<<"Enter AQI for City "<<i+1<<":\n";          for(*int* j=0;j<days;j++){              cout<<"Day "<<j+1<<":";              cin>>AQI[i][j];          }      }      calcAvg(AQI,cities,days,avg);      findBestDrop(AQI,cities);      return 0;  } |

Output



Q5

|  |
| --- |
| #include <iostream>  using *namespace* std;  *void* weeklyAvg(*int* *AQI*[4][28], *int* *cities*) {      cout<<"\nWeekly Average AQI:\n";      for(*int* i=0;i<cities;i++){          cout<<"City "<<i+1<<": ";          for(*int* w=0;w<4;w++){  *float* sum=0;              for(*int* j=w\*7;j<(w+1)\*7;j++)sum+=AQI[i][j];              cout<<"Week "<<w+1<<": "<<sum/7<<"  ";          }          cout<<endl;      }  }  *void* criticalDays(*int* *AQI*[4][28], *int* *cities*) {      cout<<"\nCritical Pollution Days (AQI > 150):\n";      for(*int* i=0;i<cities;i++){          cout<<"City "<<i+1<<": ";  *bool* found=false;          for(*int* j=0;j<28;j++){              if(AQI[i][j]>150){                  cout<<"Day "<<j+1<<"("<<AQI[i][j]<<") ";                  found=true;              }          }          if(!found)cout<<"No critical days.";          cout<<endl;      }  }  *void* minMaxAQI(*int* *AQI*[4][28], *int* *cities*) {      cout<<"\nHighest & Lowest AQI Values:\n";      for(*int* i=0;i<cities;i++){  *int* minA=999,maxA=-1;          for(*int* j=0;j<28;j++){              if(AQI[i][j]>maxA)maxA=AQI[i][j];              if(AQI[i][j]<minA)minA=AQI[i][j];          }          cout<<"City "<<i+1<<": Min "<<minA<<" | Max "<<maxA<<endl;      }  }  *int* main() {  *const* *int* cities=4,days=28;  *int* AQI[cities][days];      for(*int* i=0;i<cities;i++){          cout<<"Enter AQI for City "<<i+1<<":\n";          for(*int* j=0;j<days;j++){              cout<<"Day "<<j+1<<":";              cin>>AQI[i][j];          }      }      weeklyAvg(AQI,cities);      criticalDays(AQI,cities);      minMaxAQI(AQI,cities);      return 0;  } |

Output

