```
 1
 2 import org.json.*;
19
20
21 public class JsonParser {
22
23
24     static ArrayList<PersonItem> invited_friends= new ArrayList<PersonItem>();
25     static ArrayList<PersonItem> not_invited_friends= new ArrayList<PersonItem>();
26     static ArrayList<PersonItem> list1=new ArrayList<PersonItem>();
27     static JSONParser parser ;
28
29     public static void main(String args[]) {
30
31         //For parsing the JSON.
32          parser = new JSONParser();
33
34
35         try{ Objectparsing(); }
36
37         catch(Exception e){ e.printStackTrace(); }
38
39         //invited_friends
40         invited_friends();
41
42         //not_invited_friends
43         not_invited_friends();
44
45     System.out.println("The Friends who are in the range of 100KMS are ");
46
47     for(int i=0;i<invited_friends.size();i++)
48     {
49
50         System.out.print(invited_friends.get(i).name +" ");
51         System.out.print(invited_friends.get(i).user_id +" ");
52         System.out.print(invited_friends.get(i).latitude +" ");
53         System.out.print(invited_friends.get(i).lonitude +" ");
54         System.out.println(" ,DISTANCE IS = " + invited_friends.get(i).distance +" ");
55
56     }
57
58
59     System.out.println("\n\nThe Friends who are  not in the range of 100KMS and are not
   invited  ");
60     for(int i=0;i<not_invited_friends.size();i++)
61     {
62
63         System.out.print(not_invited_friends.get(i).name +" ");
64         System.out.print(not_invited_friends.get(i).user_id +" ");
65         System.out.print(not_invited_friends.get(i).latitude +" ");
66         System.out.print(not_invited_friends.get(i).lonitude +" ");
67         System.out.println("  ,DISTANCE IS = " + not_invited_friends.get(i).distance +" ");
68
69     }
70 }//main method ends
71
72 public static void Objectparsing() throws FileNotFoundException, IOException, ParseException
```

```
73 {
74     Object obj = parser.parse(new FileReader("Jsonfile.json"));
75     JSONArray jsonObject = (JSONArray) obj;
76 
77     for(int i =0;i<32;i++)
78     {
79         JSONObject item = (JSONObject) jsonObject.get(i);
80         String name =(String) item.get("name");
81         String lat = (String )item.get("latitude");
82         String lon = (String)item.get("longitude");
83         long id = (long)item.get("user_id");
84 
85         PersonItem personItem= new PersonItem(name,lat,lon,id);
86         list1.add(personItem);
87 
88     }
89     //given latitude and longitude
90     Double lat2 =12.9611159;
91     Double lon2 =77.6362214;
92 
93     lat2 =converting_lat_toRadians(lat2);
94     lon2 =converting_lon_toRadians(lon2);
95 
96 
97 
98     int l = list1.size();
99 
100    for(int i=0;i<l;i++)
101    {
102        Double lat1 = getting_latitude(i);
103        Double lon1 =getting_longitude(i);
104 
105        lat1 = converting_lat_toRadians(lat1);
106        lon1 =converting_lon_toRadians(lon1);
107 
108        double difference_lon = lon2 - lon1;
109        double difference_lat = lat2 - lat1;
110 
111        //using the formula.
112        double distance =calculation_of_the_distance( difference_lat, lat1, lat2,
   difference_lon);
113 
114         //method which keeps the information of those friends who are invite and are in the
   range of 100Kms.
115        calculating_the_distance(distance,i);
116    }
117 
118 }
119 //calculating the latitude
120 public  static Double getting_latitude(int i)
121    {
122 
123                //System.out.println(list1.get(i).name);
124                String lat = list1.get(i).latitude;
125                Double  lat1 = Double.parseDouble(lat);
126     return lat1;
127    }
```

```
128
129 //calculating the longitude
130 public static  Double getting_longitude(int i)
131     {
132                 String longi = list1.get(i).lonitude;
133                 Double  lon1 = Double.parseDouble(longi);
134      return lon1;
135
136     }
137
138 //converting the latitude to radians
139 public static Double converting_lat_toRadians(Double lat2)
140     {
141         lat2 = Math.toRadians(lat2);
142
143        // System.out.println(lat2);
144
145         return lat2;
146     }
147
148  //converting the longitude to radians
149 public static Double converting_lon_toRadians(Double lon2)
150     {
151             lon2 = Math.toRadians(lon2);
152          return lon2;
153     }
154
155 //calculating the distance between the latitudes and longitudes using the formula
156 public static double  calculation_of_the_distance(double difference_lat,double lat1,double
    lat2,double difference_lon)
157 {
158      double a = Math.pow(Math.sin(difference_lat / 2), 2) + Math.cos(lat1) * Math.cos(lat2)
159              * Math.pow(Math.sin(difference_lon / 2),2);
160
161      double c = 2 * Math.asin(Math.sqrt(a));
162
163      //Radius of the earth
164      double radius = 6371;
165      double distance  =radius*c;
166
167  return distance;
168
169 }
170 //calculating the distance between the given data and the friends location and if less than
    100
171 // putting the same in different Arraylist to invite them.
172 public static void calculating_the_distance(Double distance,int i)
173     {
174
175
176
177      if(distance<=100)
178                  {
179                          String name =(String)  list1.get(i).name;
180                          String lati = (String ) list1.get(i).latitude;
181                          String lon = (String) list1.get(i).lonitude;
182                          long id = (long) list1.get(i).user_id;
```

```
183
184                         PersonItem person =  new PersonItem(name,lati,lon,id,distance);
185                         invited_friends.add(person);
186                 }
187
188     else
189                  {
190                          String name =(String)  list1.get(i).name;
191                          String lati = (String ) list1.get(i).latitude;
192                          String lon = (String) list1.get(i).lonitude;
193                          long id = (long) list1.get(i).user_id;
194
195                         PersonItem person =  new PersonItem(name,lati,lon,id,distance);
196                         not_invited_friends.add(person);
197                 }
198     }
199
200 //invited_friends
201 public static void invited_friends()
202 {
203       //sorting the PersonItem on the basis of user_id .
204     class Sortbyroll implements Comparator<PersonItem>
205     {
206         public int compare(PersonItem a, PersonItem b)
207         {
208             return (int) (a.user_id - b.user_id);
209         }
210
211     }
212
213 //sorting  function used here.
214 Collections.sort(invited_friends, new Sortbyroll());
215 }
216
217 //not_invited_friends
218 public static void not_invited_friends()
219 {
220      class Sortbyroll1 implements Comparator<PersonItem>
221         {
222             public int compare(PersonItem a, PersonItem b)
223             {
224                 return (int) (a.user_id - b.user_id);
225             }
226
227         }
228
229     //sorting  function used here.
230     Collections.sort(not_invited_friends, new Sortbyroll1());
231 }
232
233
234 }//class ends
235
236
237 //output
238 /*
239  The Friends who are in the range of 100KMS are
```

```
Ian 4 13.2411022 77.238335  ,DISTANCE IS = 53.161349697383876
Nora 5 13.1302756 77.2397222  ,DISTANCE IS = 46.88894185833014
Theresa 6 13.1229599 77.2701202  ,DISTANCE IS = 43.55066151846466
Georgina 10 12.240382 77.972413  ,DISTANCE IS = 88.05496818333745
Richard 11 13.008769 77.1056711  ,DISTANCE IS = 57.72964164658059
Chris 12 12.986375 77.043701  ,DISTANCE IS = 64.26480291997055
Michael 15 12.966 77.463  ,DISTANCE IS = 18.77828059115435
Ian 16 12.366037 78.179118  ,DISTANCE IS = 88.5859736791806
David 25 12.833502 78.122366  ,DISTANCE IS = 54.57024856881082
Alan 31 13.1489345 77.8422408  ,DISTANCE IS = 30.56426320779474
Lisa 39 13.0033946 77.3877505  ,DISTANCE IS = 27.32987841081551


The Friends who are  not in the range of 100KMS and are not invited
Alice 1 11.92893 78.27699   ,DISTANCE IS = 134.21516365357343
Ian 2 11.8856167 78.4240911   ,DISTANCE IS = 147.04189090209707
Jack 3 12.3191841 78.5072391   ,DISTANCE IS = 118.43223818016337
Frank 7 13.4692815 -9.436036   ,DISTANCE IS = 9365.223531002577
Eoin 8 14.0894797 77.18671   ,DISTANCE IS = 134.55072657121264
Jack 9 12.2559432 76.1048927   ,DISTANCE IS = 183.7396470014124
Olive 13 13 76   ,DISTANCE IS = 177.34270530574926
Helen 14 11.999447 -9.742744   ,DISTANCE IS = 9431.96122233646
Patricia 17 14.180238 -5.920898   ,DISTANCE IS = 8977.543796490403
Bob 18 12.228056 76.915833   ,DISTANCE IS = 112.94107764995155
Enid 19 55.033 78.112   ,DISTANCE IS = 4678.36314726356
Enid 20 13.121111 -9.831111   ,DISTANCE IS = 9415.111135889154
David 21 11.802 -9.442   ,DISTANCE IS = 9404.59800382023
Charlie 22 14.374208 78.371639   ,DISTANCE IS = 176.0760647683372
Eoin 23 14.080556 77.361944   ,DISTANCE IS = 127.95919922148792
Rose 24 14.133333 77.433333   ,DISTANCE IS = 132.1768775680275
Stephen 26 13.038056 76.613889   ,DISTANCE IS = 111.09462442885558
Enid 27 14.1225 78.143333   ,DISTANCE IS = 140.29369783566008
Charlie 28 13.807778 76.714444   ,DISTANCE IS = 137.13371823622575
Oliver 29 13.74412 76.11167   ,DISTANCE IS = 186.5074584995188
Nick 30 13.761389 76.2875   ,DISTANCE IS = 170.9043788302323
*/

```