**package** DivideAndConquer;

**import** java.util.ArrayList;

**import** java.util.Collections;

**class** Activity **implements** Comparable<Activity>

{

**public int activityName**;

**public int startTime**;

**public int endTime**;

**public** Activity(**int** activityName, **int** startTime, **int** endTime) {

**this**.**activityName** = activityName;

**this**.**startTime** = startTime;

**this**.**endTime** = endTime;

}

@Override

**public int** compareTo(Activity o) {

**if** (**this**.**endTime**>o.**endTime**) **return** 1;

**else return** -1;

}

}

**class** ActivitySelectionThere

{

ArrayList<Activity> **activities**;

**public int**[] **arary**;

**public int index** =0;

**int count** =0;

**public int firstEndTime**;

**public** ActivitySelectionThere(ArrayList<Activity> activities) {

**this**.**activities** = activities;

**this**.**arary** = **new int**[100];

**this**.**firstEndTime** =0;

}

**public void** giveData()

{

**for** (Activity give : **activities**)

{

**if** (**count**==0) {

System.***out***.println(**"Activity Name: "**+give.**activityName** + **" "**);

compare(give.**activityName**, give.**startTime**, give.**endTime**,**count**);

**count**++;

}**else** {

compare(give.**activityName**, give.**startTime**, give.**endTime**,**count**);

}

}

}

**public void** compare(**int** activityName, **int** startTime, **int** finishTime,**int** count)

{

**if** (count ==0)

{

**firstEndTime** = finishTime;

}**else**

{

**if** (**firstEndTime**<startTime)

{

**arary**[**index**] = activityName;

**firstEndTime** = finishTime;

**index** ++;

**return**;

}

**else** {}

}

}

**public void** print()

{

**for** (**int** i=0;i<**index**;i++)

{

System.***out***.println(**"Activity Name: "**+**arary**[i]+**" "**);

}

}

}

**public class** ActivitySelection {

**public static void** main(String[] args) {

ArrayList<Activity> activities = **new** ArrayList<>();

activities.add(**new** Activity(1,1,3));

activities.add(**new** Activity(2,2,5));

activities.add(**new** Activity(3,4,7));

activities.add(**new** Activity(4,1,8));

activities.add(**new** Activity(5,5,9));

activities.add(**new** Activity(6,8,10));

activities.add(**new** Activity(7,9,11));

activities.add(**new** Activity(8,11,14));

activities.add(**new** Activity(9,13,16));

Collections.*sort*(activities);

ActivitySelectionThere object = **new** ActivitySelectionThere(activities);

object.giveData();

object.print();

}

}