Program 1:

**package** com.holidify;

//Program 1

**public** **class** A {

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

**int** a[]=**new** **int**[] {5,10,3};

**int** a1[]=**new** **int**[] {5,10,20};

**int** sum=0,sum1=0,max=0,max1=0;

**for**(**int** i=0;i<a.length;i++) {

sum=sum+a[i];

}

**for**(**int** i=0;i<a.length;i++) {

**if**(a[0]<a[i]) {

max=a[i];

}

}

**int** total=sum-max;

System.***out***.println(total);

**for**(**int** i=0;i<a1.length;i++){

sum1=sum1+a1[i];

}

**for**(**int** i=0;i<a1.length;i++) {

**if**(a[0]<a1[i]) {

max1=a1[i];

}

}

**int** total1=sum1-max1;

System.***out***.println(total1);

}

}

Program 2:

**package** com.holidify;

**public** **class** B {

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

{

String a = "1101001", b="100101";

System.***out***.print(*addBinary*(a, b));

}

**static** String addBinary(String a, String b)

{

**if**(a.charAt(0) == '0' && b.charAt(0) == '0'){

**return** "0";

}

StringBuilder result = **new** StringBuilder("");

**int** s = 0;

**int** i = a.length() - 1, j = b.length() - 1;

**while** (i >= 0 || j >= 0 || s == 1)

{

s += ((i >= 0)? a.charAt(i) - '0': 0);

s += ((j >= 0)? b.charAt(j) - '0': 0);

result.append((**char**)(s % 2 + '0'));

s /= 2;

i--; j--;

}

**int** start = result.length()-1;

**while**(start >=0 && result.charAt(start) == '0') {

start--;

}

**if**(start != result.length()-1) {

result.delete(start+1,result.length());

}

**return** result.reverse().toString();

}

}

Program 3:

**package** com.holidify;

**import** java.util.HashMap;

**import** java.util.Map;

**public** **class** D

{

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

    {

        Map<String, String> map = **new** HashMap<String, String>();

        map.put("Mangalore", "Banglore");

        map.put("Bombay", "Mysore");

        map.put("Surathkal", "Mangalore");

        map.put("Mysore", "Surathkal");

        printResult(map);

    }

**private** **static** **void** printResult(Map<String, String> map)

    {

        Map<String, String> reverseMap = **new** HashMap<String, String>();

**for** (Map.Entry<String,String> entry: map.entrySet())

            reverseMap.put(entry.getValue(), entry.getKey());

        String start = **null**;

**for** (Map.Entry<String,String> entry: map.entrySet())

        {

**if** (!reverseMap.containsKey(entry.getKey()))

              {

                   start = entry.getKey();

**break**;

              }

        }

**if** (start == **null**)

        {

           System.out.println("Invalid Input");

**return**;

        }

        String to = map.get(start);

**while** (to != **null**)

        {

            System.out.print(start + "->" + to + ", ");

            start = to;

            to = map.get(to);

        }

    }

}

Database:

4. SELECT name, AVG(rating) AS AVGRating

FROM attractions as a

INNER JOIN destinantions as r

ON a.ID = r.attractions\_ID

GROUP BY name

ORDER BY AVGRating DESC;