```
/**
 * methods for retrieving a range of rows of weather station observations from an HBase table
 * here we are interested in a Map<ObservationTime, ObservedTemp> result
 * we will use a NavigableMap<Long, Integer> because it is sorted and has a descendingMap()
method
 * so we can access observations in both ascending and descending order
 */
public NavigableMap<Long, Integer> getStationObservations(HTable, table, String stationID,
long maxStamp, int maxCount) throws IOException {
        byte[][] columns = {Bytes.toBytes("data:airtemp")};
        byte[] startRow = RowKeyConverter.makeObservationRowKey(stationId, maxStamp);
        RowResult res = null;
        NavigableMap<Long, Integer> resultMap = new TreeMap<Long, Integer>();
        byte[] airtempColumn = Bytes.toBytes("data:airtemp");
        // scanners are like cursors in a traditional database. scanners return rows in order
        Scanner s = table.getScanner(columns, startRow);
        int count = 0;
        try {
                while ((res = s.next()) != null && count++ < maxCount) {
                        byte[] row = res.getRow();
                        byte[] value = res.get(airtempColumn).getValue();
                        Long stamp = Long.MAX_VALUE - Bytes.toLong(row, row.length -
Bytes.SIZEOF_LONG, Bytes.SIZEOF_LONG);
                        Integer temp = Bytes.toInt(value);
                        resultMap.put(stamp, temp);
                }
        } finally {
                s.close();
        }
        return resultMap;

        /**
         * Return the last ten observations
         */
        public NavigableMap<Long, Integer> getStationObservations(HTable table, String
stationId) throws IOException {
                return getStationObservations(table, stationId, Long.MAX_VALUE, 10);
        }
```