Question 1:

Use secondary sort.
Natural Key: userID; Secondary Key: price; Value: product

Class Mapper<Text, Text, Pair, Text>
  method Map(String t)
    if(time.contains('2016')):
      Emit(Pair(String t.userID, Int(t.price)), t.product)
Class PairPartitioner
  method Partition
    return Pair.userID //pairs associate with the same userID are shuffled to the same reducer
Class KeyGroupComparator
  method GroupKey
    compare(Pair1.userID, Pair2.userID)
  method ComparePrice
    compair(Pair1.price, Pair2.price)
Class Reducer<Pair, Text, Text, Text>
  method Reduce
    for pair <— 0 to 4
      Emit(pair.userID, products)

Question 2:

Row	C1	C2	h1	h2
0	0	1	2	6
1	1	0	5	1
2	0	1	1	3
3	0	0	4	5
4	1	1	0	0
5	1	1	3	2
6	1	0	6	4

Initialize:

	C1	C2
h1	∞	∞
h2	∞	∞

Row 0:

	C1	C2
h1	∞	**2**
h2	∞	**6**

Row 1:

	C1	C2
h1	**5**	2
h2	**1**	6

Row 2:

	C1	C2
h1	5	**1**
h2	1	**3**

Row 3:

	C1	C2
h1	5	1
h2	1	3

Row 4:

	C1	C2
h1	**0**	**0**
h2	**0**	**0**

Row 5:

	C1	C2
h1	0	0
h2	0	0

Row 6:

	C1	C2
h1	0	0
h2	0	0

Result :

Signature for C1:

0	0

Signature for C2:

0	0

Question 3:

initial:
(16, 148)(8,162)(8,177)(4,183)(2,192)(1,197)(1,200)

time 201:
current bit is 0, no other changes needed.

time 202:
(16, 148)(8,162)(8,177)(4,183)(2,192)(1,197)(1,200)(1,202)

time 203:
current bit is 0, no other changes needed

time 204:
(16, 148)(8,162)(8,177)(4,183)(2,192)**(1,197)(1,200)**(1,202)(1,204)
(merge)
=> (16, 148)(8,162)(8,177)(4,183)(2,192)(2,200)(1,202)(1,204)

time 205:
current bit is 0, no other changes needed

time 206:
(16, 148)(8,162)(8,177)(4,183)(2,192)(2,200)**(1,202)(1,204)**(1,206)
(merge)
=> (16, 148)(8,162)(8,177)(4,183)**(2,192)(2,200)**(2,204)(1,206)
(merge)
=> (16, 148)(8,162)(8,177)(4,183)(4,200)(2,204)(1,206)

time 207:
current bit is 0, no other changes needed

time 208:
(16, 148)(8,162)(8,177)(4,183)(4,200)(2,204)(1,206)(1,208)

time 209:
current bit is 0, no other changes needed

time 210:
(16, 148)(8,162)(8,177)(4,183)(4,200)(2,204)**(1,206)(1,208)**(1,210)
(merge)
=> (16, 148)(8,162)(8,177)(4,183)(4,200)(2,204)(2,208)(1,210)

current time = 210, window size = 60, 210 - 60 = 150 > 148(first timestamp)

=> **result: (8,162)(8,177)(4,183)(4,200)(2,204)(2,208)(1,210)**

Question 4:

$$sim(x,y) = \frac{\sum_i r_{xi} * r_{yi}}{\sqrt{\sum_i r_{xi}^2} * \sqrt{\sum_i r_{yi}^2}} \qquad r_{xi} = \frac{\sum_{j \in (x;i)} S_{ij} * r_{jx}}{\sum S_{ij}}$$

(a) **user - user CF**

	m1	m2	m3
u1	2		3
u2	5	2	
u3	3	3	1
u4		2	2

sim(u1, u2) = 0.515
sim(u1, u3) = 0.573
sim(u1, u4) = 0.588

Rating of u1 to m2: (0.515*2 + 0.573*3 + 0.588*2)/(0.515+0.573+0.588) = 2.34

(b) **item - item CF**

	u1	u2	u3	u4
m1	2	5	3	
m2		2	3	2
m3	3		1	2

sim(m2, m1) = 0.748
sim(m2, m3) = 0.454

Rating of u1 to m2: (0.748*2 + 0.454*3)/(0.748 + 0.454) = 2.38