1.For each peer_id, get the year when peer_id contains id_2, for example for ‘ABC17969(AB)’ year is 2022.

```
package Scala
import scala.io.Source
object second_code {
    def main(args: Array[String]): Unit = {
        case class Person(peer_id: String, id_1: String, id_2: String, year:Int)
        val people = List(
            Person("ABC17969(AB)","1", "ABC17969", 2022),
            Person("ABC17969(AB)", "2", "CDC52533", 2022),
            Person("AE686(AE)", "7", "AE686", 2023))
        val namesStartingWithA = for {
            person <- people
            if person.id_2.startsWith("ABC17969")
            name = person.peer_id+"    year is    "+person.year
        } yield name
        println(namesStartingWithA.mkString(","))
        val namesStartingWith_other = for {
            person <- people
            if person.id_2.startsWith("AE686")
            name1 = person.peer_id+"    year is    "+person.year
        } yield name1
        println(namesStartingWith_other.mkString(","))
    }
}
```

2 Given a size number, for example 3. For each peer_id count the number of each year

```
package Scala
import scala.io.Source
import scala.collection.mutable.HashMap
object code {
        def main(args: Array[String]): Unit = {
            case class Person(peer_id: String, id_1: String, id_2: String, year:Int)
            val people = List(
                Person("ABC17969(AB)","1", "ABC17969", 2022),
                Person("ABC17969(AB)", "2", "CDC52533", 2022),
                Person("ABC17969(AB)", "3", "DEC59161", 2023),
            Person("ABC17969(AB)", "4", "F43874", 2022),
            Person("ABC17969(AB)", "5", "MY06154", 2021),
            Person("ABC17969(AB)", "6", "MY4387", 2022)
            )
            val    group =people.groupBy(_.year)
              val sc=group.map(x=>x._1 ->x._2.size)
```

```
          println(sc)
    }

}
```

3.Order the value in step 2 by year and check if the count number of the first year is bigger or equal than the given size number. If yes, just return the year.
If not, plus the count number from the biggest year to next year until the count number is bigger or equal than the given number. For example, for ‘AE686(AE)’, the year is 2023, and count are:

```
package Scala
import scala.io.Source
import scala.collection.mutable.HashMap
object code {

    def main(args: Array[String]): Unit = {

      case class Person(peer_id: String, id_1: String, id_2: String, year:Int)
      val people = List(
        Person("ABC17969(AB)","1", "ABC17969", 2022),
        Person("ABC17969(AB)", "2", "CDC52533", 2022),
        Person("ABC17969(AB)", "3", "DEC59161", 2023),
      Person("ABC17969(AB)", "4", "F43874", 2022),
      Person("ABC17969(AB)", "5", "MY06154", 2021),
      Person("ABC17969(AB)", "6", "MY4387", 2022)
      )
      val    group =people.groupBy(_.year)
       val sc=group.map(x=>x._1 ->x._2.size)
          println(sc)

      val inputInt = scala.io.StdIn.readInt()
      //val inputString = scala.io.StdIn.readLine()

      val hashMap = HashMap(2023 -> 1, 2022 -> 4, 2021 -> 1)



      for(k <- hashMap.keySet) {
        println(s"value:${hashMap(k)}")
        val integerHash = Integer.toUnsignedLong(hashMap(k))
        if (integerHash=inputInt){
          println(s"peer_id $year")

        }else if (hashMap(k)+tail.hasMap(k)>=inputInt)
          println(s"$peer_id $year")
```

```
            }





        }

    }
```