实操一

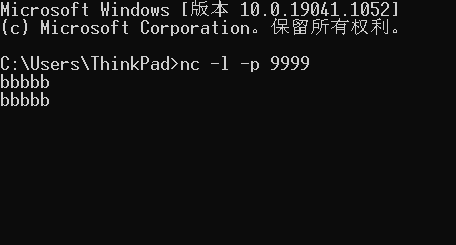
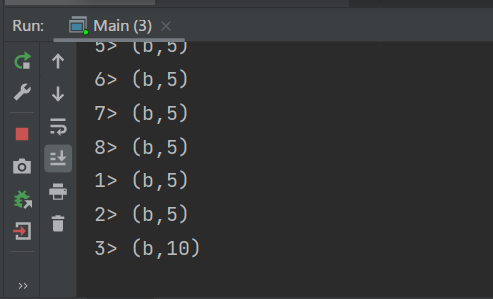
**输出1分钟内字符’b’的次数：**

**关键代码：**引入Time依赖，并使用split("")来分割单个字符，最后用timeWindowAll(Time.minutes(1), Time.seconds(1))函数来创建滑动窗口，其中窗口长度为1分钟，每间隔1秒钟刷新窗口，并用sum(1)统计字符串出现次数。

import org.apache.flink.streaming.api.windowing.time.Time

val text: DataStream[String] = env.socketTextStream("localhost", 9999)  
// val stream = text.timeWindowAll(Time.minutes(1))  
val stream: DataStream[(String, Int)] = text.flatMap(x => x.toLowerCase.split("").filter(x => x.contains(*target*)))  
 .map(x => (x, 1))  
 .keyBy(0)  
 .timeWindowAll(Time.*minutes*(1), Time.*seconds*(1))  
 .sum(1)

**结果：**

实操二

**为什么返回的字段没有按时间排序？**

Kafka的数据存储在不同的partition，Kafka在返回数据时是从多个partition同时返回，因此数据不一定有序，如果指定从某一个partition返回数据，则数据是有序的，因为Kafka保证在每个partition内部数据是有序的。

**统计乘客到达数前五的城市：**

**关键代码：先根据关键字“destination”进行词频统计并根据词频排序，然后创建一个1分钟的滑动窗口，每隔10s刷新一次，输出排序结果。**

inputKafkaStream.map(x => x.substring(x.indexOf("destination")+14,x.indexOf("username")-3))  
 .map(x => (x, 1))  
 .keyBy(0)  
 .window(SlidingProcessingTimeWindows.*of*(Time.*seconds*(60L), Time.*seconds*(10L)))  
 .sum(1)  
 .windowAll(TumblingProcessingTimeWindows.*of*(Time.*seconds*(10L)))  
 .process(new ProcessAllWindowFunction[(String, Int), String, TimeWindow] {  
 override def process(context: Context, elements: Iterable[(String, Int)], out: Collector[String]): Unit = {  
 val top3 = elements.toSeq  
 .sortBy(-\_.\_2)  
 .take(5)  
 .zipWithIndex  
 .map { case ((item, price), idx) => s" **$**{idx + 1}. **$**item: **$**price" }  
 .mkString("\n")  
 out.collect(("-" \* 16) + "\n" + top3)  
 }  
 }  
 )

**结果：**



实操三

**读取mysql的数据如kafka流：**

**关键代码：使用kafka的python接口，先连接到kafka，然后把读取到的mysql数据库内容发送到kafka的数据库里，其中TOPIC为lcx\_1。**

producer = KafkaProducer(*bootstrap\_servers*=config.SERVER,

*value\_serializer*=lambda *m*: json.dumps(m,*ensure\_ascii*=True).encode())

for i in mysql\_data.getdata():

    producer.send(config.TOPIC, i)

    time.sleep(1)

**结果：左边是加入后的内容，右边是数据库里的内容。**



实操四

**将flink结果写入mysql：**

**关键代码：创建jdbc mysql数据库连接，然后将结果写入mysql。**

val *url* = "jdbc:mysql://localhost:3306/mysql?serverTimezone=GMT"  
//驱动名称  
val *driver* = "com.mysql.jdbc.Driver"

def getConnection(): Connection = {  
 DriverManager.*getConnection*(*url*, *username*, *password*)  
}

实操五

**关键字匹配实施输入：**

**关键代码：将关键字获得改为从控制台手动输入。**

//需要监控的人名  
*println*("Enter the key: ")  
val *user* = Console.readLine()

**结果：输入关键字“小学生”，得到查询结果：**

