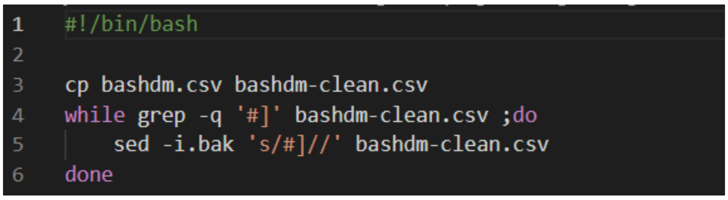
In this report, I will shortly and roughly introduce my work during the process of doing assignment2. Some rationales of code and how did I design and come up with that method will be illustrated as well.

In order to run the whole program, please use the code of README in every dictionary.

```Declaration of some problems I have in my submission```
The first thing is about that I had modified the data file of Spark part since I have no idea why the "No.Reviews" cannot be recognized, so I changed it to "Reviews". But I found a method to address this problem which is using `No.Reviews` instead.
The second thing is that I also had modified the data file of GraphX part, because when I was trying to create graph, I had to locate the id number of origin and destination, but when I was coping with that thing, I found there are some places didn't have id number in the vertices data file. So, I just deleted some data to make sure they all were able to match their id number. I also submitted my enquiry on ticket system, but it seems that hasn't been replied until now.

THE FIST PART ---- Clean-up Tasks
Q1:

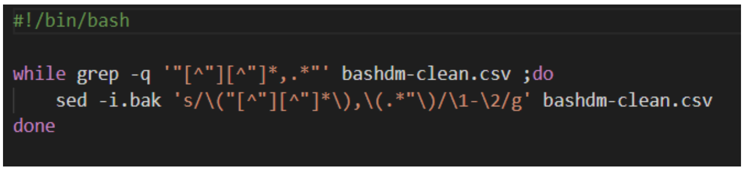


```
#!/bin/bash

cp bashdm.csv bashdm-clean.csv
while grep -q '#]' bashdm-clean.csv ;do
    sed -i.bak 's/#]//' bashdm-clean.csv
done
```

I firstly copy data file, use where command to iterate csv file and grep to select the sentence with "#]" and finally use sed command to replace them with "".

Q2:

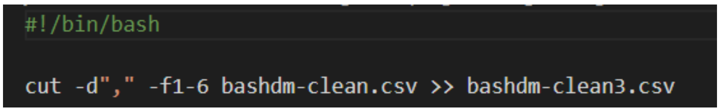


```
#!/bin/bash

while grep -q '"[^"][^"]*,.*"' bashdm-clean.csv ;do
    sed -i.bak 's/\("[^"][^"]*\),\(.*"\)/\1-\2/g' bashdm-clean.csv
done
```

In term of question2, I initially change all the "-" symbol to ",", and determine the "," inside of ' " ' (double quotation mark), and change them back to "-".

Q3:

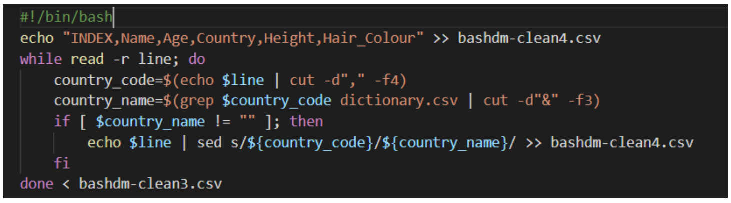


```
#!/bin/bash

cut -d"," -f1-6 bashdm-clean.csv >> bashdm-clean3.csv
```

Using cut command to select content from index1 to index6 which is cut by ","

Q4:



```bash
#!/bin/bash
echo "INDEX,Name,Age,Country,Height,Hair_Colour" >> bashdm-clean4.csv
while read -r line; do
    country_code=$(echo $line | cut -d"," -f4)
    country_name=$(grep $country_code dictionary.csv | cut -d"&" -f3)
    if [ $country_name != "" ]; then
        echo $line | sed s/${country_code}/${country_name}/ >> bashdm-clean4.csv
    fi
done < bashdm-clean3.csv
```

Extracting the countries’ code and name separately from data file and using sed to replace code with name and save the new line to csv file.

THE SECOND PART ---- Data management Tasks

Q1:

mysql -h localhost -D $DB -e "insert into bashdm (name, age, country, height, hair_colour) values('$name', $age, '$country', $height, '$hair')"

This is partial mian method to store data into mysql database, I use non-interactive model command to insert every row data into bashdm table.

mongo firstdatabase --eval 'db.bashdm.insert({id:"'"$a"'", number: "'"$number"'", name: "'"$name"'", age: "'"$age"'", country: "'"$country"'", height: "'"$height"'", hair_colour: "'"$hair_colour"'" })'   a=$[$a+1]

This is partial mian method to store data into mongo database, I use non-interactive model command to insert every row data into bashdm table. And this also includes question4 I use a variable as value of a new attribute called id.

Q2:

select country, avg(bashdm.height) from bashdm group by country;

Show the average height per country.

Q3:

Select hair_colour, max(bashdm.height) from bashdm group by hair_colour;

Show the maximum height per hair colour.

Q4: showed in the Q1 part

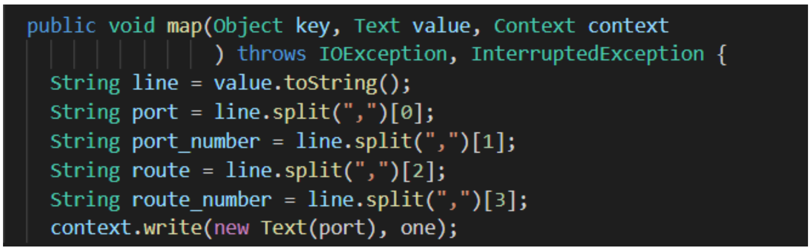
Q5:

db.bashdm.find().sort({“height”:1}).limit(1)

using sort method to find the smallest height and limit to show the first one of sort result.

THE THIRD PART ---- Simple Hadoop Graph Processing

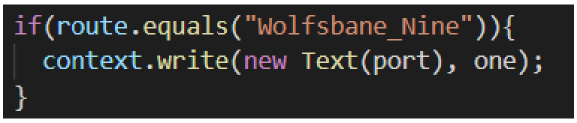
Q1:



```java
public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException {
  String line = value.toString();
  String port = line.split(",")[0];
  String port_number = line.split(",")[1];
  String route = line.split(",")[2];
  String route_number = line.split(",")[3];
  context.write(new Text(port), one);
```

I modify map function and extract port name and pass it as Text as well as an IntWritable variable to reduce.

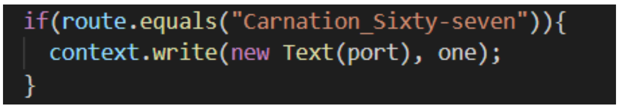
Q2:



```java
if(route.equals("Wolfsbane_Nine")){
  context.write(new Text(port), one);
}
```

The key part is using a if conditional statement to only pass port associated with a specific route name.

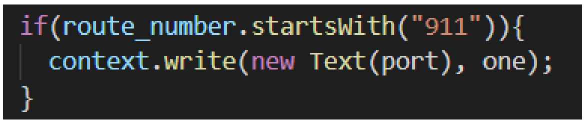
Q3:



```java
if(route.equals("Carnation_Sixty-seven")){
  context.write(new Text(port), one);
}
```

It is like Q2, only pass port names to reduce associated with a specific route name.

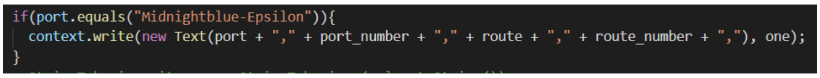
Q4:



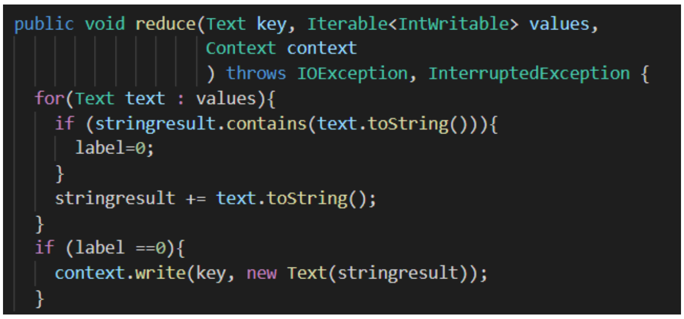
```java
if(route_number.startsWith("911")){
  context.write(new Text(port), one);
}
```

Only pass port names to reduce associated with route number which starts with “991” instead of containing “911”.

Q5:



```java
if(port.equals("Midnightblue-Epsilon")){
  context.write(new Text(port + "," + port_number + "," + route + "," + route_number + ","), one);
}
```



```java
public void reduce(Text key, Iterable<IntWritable> values,
                   Context context
                   ) throws IOException, InterruptedException {
  for(Text text : values){
    if (stringresult.contains(text.toString())){
      label=0;
    }
    stringresult += text.toString();
  }
  if (label ==0){
    context.write(key, new Text(stringresult));
  }
```

As for Q5, I firstly output the content which contains this specific port name into data file, and another mapreduce program reads both of these file and output port names which contains the route name which previous port contains.

THE FOURTH PART ---- Spark

Q1:

```
val inputFile = sc.textFile("/spark.csv")
inputFile.count
```

The count method can return how many rows there are.

Q2:

```
val data = spark.sql("select max(Reviews) Reviews from res_table").collect
val data = spark.sql("select Restaurant from res_table where Reviews=1500").collect
```

After uploading data file and create a table on the dataset, I use sql command to return the max number of reviews and get the restaurant name which has the maximum of reviews.

The alternative way to solve this problem is showed below:

```
info_res.where(info_res("_c4") === info_res.agg(max("_c4")).first()(0)).show()
```

Q3:

```
spark.sql("select Restaurant, length(Restaurant) from res_table where length(restaurant)=(select max(length(Restaurant)) from res_table)").collect.foreach(println)
```

As for question3, I also use sql command to show the restaurant name and its name’s length where the length is max.

Q4:

```
val question4=spark.sql("select Region,sum(Reviews) from res_table group by Region order by 2").collect.foreach(println)
```

Question4’s solution is using sql command to show sum of reviews of different types of region. And collect.foreach(println) is the way to print each row information.

Q5:

```
inputFile.flatMap(line => line.split(",")(4).split(" ")).map(word => {if(word!="A"&&word!="The"&&word!="of") (word,1) else (word,0)}).reduceByKey( _ +_).sort
```

To address question5 problem, I use mapreduce function to calculate the most frequent word which cannot be “A”, “The” and “of”. Before doing that, I use split function to extract the content of review from row of data.

THE FIFTH PART ---- GraphX

Notice: the process of completing question1 is quite complicated, and the detailed process will be showed inside GraphX dictionary. Right here, I will introduce the idea of how I try to address these problems.

Q1:
In order to create a graph. We have to make sure we already have vertices and edges. Int this case, vertices concludes the id number of harbour and harbours’ name. And edges conclude the origin’s id number ,destination’s id number and a value which could be the distance or other value. But there is a problem that in the edge data file, we only have the name of origin and destination, so that I decided to extract the id number and harbours’ name into two list. And when I were trying to create edges, I can use id number of list to replace origins and destinations’ name.
The detailed information and process of question1, please see the pdf inside of GrpahX dictionary and README file.

Q2:

```
val direction: EdgeDirection = EdgeDirection.Either
graph.collectEdges(direction).collect()
```

I learnt this method from the GraphOps documentation, and this methos can return an RDD that contains for each vertex v its local edges. And direction.either is aimed to get all the edges regardless of its direction.

Q3:

```
graph.edges.filter{case (Edge(org_id, dset_id,
route_name))=>route_name=="Heather_Three_hundred_and_eighty-four"}.take(3)
```

This method is aimed to return the edges which route name is specific a name. I tried to show the first three of result. However, there is only one result.

Q4:

```
def max(a:(VertexId, Int), b:(VertexId, Int)):(VertexId, Int) = {if(a._2>b._2) a else b}
val maxDegrees: (VertexId, Int) = graph.degrees.reduce(max)
```

I learnt this from the previous lab, and it shows how many routes a harbour connected and will return the max one.

Q5:

```
graph.collectNeighborIds(EdgeDirection.Either).collect.foreach(n=>println((n._1)+ "'s
neighbours:"+ n._2.distinct.mkString(",")))
val question5 = graph.collectNeighborIds(EdgeDirection.Either)
question5.collect
val result = question5.collect.sortBy(r => (r._2.length,
r._1.toInt))(Ordering.Tuple2(Ordering.Int.reverse, Ordering.Int.reverse))
```

In terms of question5, I used the collectNeighborIds method to get all harbours connected by a certain harbour. And I also used sort function to sort this RDD based on number of how many harbours a certain harbour has connected.