实验四：RDD编程初级实践

一、实验目的

- 熟悉 `Spark` 的 `RDD` 基本操作及键值对操作；
- 熟悉使用 `RDD` 编程解决实际具体问题的方法。

二、实验平台

操作系统： `Ubuntu16.04`

`Spark` 版本：2.1.0

1. `spark-shell` 交互式编程

请在班级群里下载 `Data01.txt` (下载)，该数据集包含了某大学计算机系的成绩，数据格式如下所示：

```
Aaron,OperatingSystem,100
Aaron,Python,50
Aaron,ComputerNetwork,30
Aaron,Software,94
Abbott,DataBase,18
Abbott,Python,82
Abbott,ComputerNetwork,76
Abel,Algorithm,30
Abel,DataStructure,38
Abel,OperatingSystem,38
Abel,ComputerNetwork,92
Abraham,DataStructure,12
Abraham,ComputerNetwork,78
Abraham,Software,98
Adair,DataBase,20
Adair,Python,98
Adair,Software,88
Adam,Algorithm,18
Adam,ComputerNetwork,70
Adam,Software,80
Adolph,DataStructure,82
Adolph,CLanguage,100
Adolph,ComputerNetwork,70
Adolph,Software,18
Adonis,DataBase,86
Adonis,Algorithm,34
Adonis,DataStructure,52
Adonis,CLanguage,30
Adonis,Python,86
Alan,Algorithm,48
Alan,OperatingSystem,86
Alan,CLanguage,72
Alan,Python,94
Alan,ComputerNetwork,88
Albert,DataStructure,60
Albert,CLanguage,76
```

```
Albert,ComputerNetwork,62
Aldrich,DataBase,42
Aldrich,Python,98
Aldrich,ComputerNetwork,80
Alexander,Algorithm,56
Alexander,DataStructure,4
Alexander,CLanguage,74
Alexander,Python,70
Alfred,Algorithm,60
Alfred,Python,96
Alger,Algorithm,50
Alger,OperatingSystem,32
Alger,Python,96
Alger,ComputerNetwork,20
Alger,Software,74
Allen,Algorithm,76
Allen,OperatingSystem,70
Allen,Python,10
Allen,Software,76
Alston,Algorithm,78
Alston,DataStructure,74
Alston,Python,96
Alston,Software,28
Alva,DataBase,72
Alva,DataStructure,64
Alva,CLanguage,0
Alva,ComputerNetwork,58
Alva,Software,82
Alvin,DataBase,88
Alvin,Algorithm,96
Alvin,OperatingSystem,26
Alvin,Python,84
Alvin,ComputerNetwork,76
Alvis,Algorithm,18
Alvis,DataStructure,56
Alvis,OperatingSystem,64
Alvis,CLanguage,56
Alvis,Python,64
Alvis,ComputerNetwork,56
Amos,DataBase,60
Amos,Algorithm,22
Amos,DataStructure,46
Amos,OperatingSystem,42
Amos,ComputerNetwork,4
Andrew,Algorithm,96
Andrew,DataStructure,62
Andrew,CLanguage,20
Andrew,Python,94
Andy,Algorithm,52
Andy,Python,76
Andy,ComputerNetwork,20
Angelo,CLanguage,30
Angelo,Software,54
Antony,DataBase,100
Antony,OperatingSystem,72
Antony,CLanguage,98
```

```
Antony,Python,46
Antony,ComputerNetwork,28
Antonio,DataBase,92
Antonio,CLanguage,22
Antonio,ComputerNetwork,0
Archer,Algorithm,18
Archer,OperatingSystem,70
Archer,CLanguage,44
Archer,Python,54
Archer,Software,10
Archibald,DataBase,20
Archibald,Algorithm,0
Archibald,CLanguage,30
Archibald,Python,84
Archibald,ComputerNetwork,30
Aries,Algorithm,60
Aries,DataStructure,10
Arlen,DataStructure,34
Arlen,OperatingSystem,2
Arlen,ComputerNetwork,52
Arlen,Software,54
Armand,DataBase,26
Armand,DataStructure,42
Armand,OperatingSystem,18
Armstrong,DataBase,28
Armstrong,Software,26
Baron,Algorithm,12
Baron,DataStructure,40
Baron,OperatingSystem,72
Baron,CLanguage,86
Baron,ComputerNetwork,96
Baron,Software,54
Barry,DataStructure,90
Barry,OperatingSystem,60
Barry,Python,100
Barry,ComputerNetwork,28
Barry,Software,16
Bartholomew,Algorithm,16
Bartholomew,CLanguage,44
Bartholomew,Python,100
Bartholomew,ComputerNetwork,34
Bartholomew,Software,50
Bart,DataBase,64
Bart,Algorithm,12
Bart,DataStructure,62
Bart,Python,56
Bart,Software,8
Barton,Python,90
Basil,DataBase,8
Basil,CLanguage,92
Basil,Python,98
Basil,Software,48
Beck,DataBase,92
Beck,DataStructure,66
Beck,OperatingSystem,30
Beck,ComputerNetwork,0
```

```
Ben,DataBase,52
Ben,Algorithm,100
Ben,Python,40
Ben,ComputerNetwork,42
Benedict,DataBase,60
Benedict,DataStructure,96
Benedict,CLanguage,8
Benedict,Python,98
Benedict,ComputerNetwork,84
Benedict,Software,76
Benjamin,Algorithm,74
Benjamin,DataStructure,94
Benjamin,Python,60
Benjamin,Software,82
Bennett,DataBase,88
Bennett,Algorithm,42
Bennett,DataStructure,60
Bennett,CLanguage,74
Bennett,ComputerNetwork,56
Bennett,Software,38
Benson,Algorithm,64
Benson,DataStructure,52
Benson,OperatingSystem,38
Benson,CLanguage,86
Berg,Algorithm,88
Berg,DataStructure,28
Berg,CLanguage,92
Berg,Python,70
Bernard,DataStructure,46
Bernard,Python,98
Bernie,DataStructure,46
Bernie,ComputerNetwork,4
Bernie,Software,28
Bert,DataBase,58
Bert,Python,16
Bert,Software,94
Bertram,OperatingSystem,54
Bertram,ComputerNetwork,86
Bertram,Software,4
Bevis,OperatingSystem,74
Bevis,CLanguage,66
Bevis,Python,84
Bevis,ComputerNetwork,72
Bill,DataBase,56
Bill,ComputerNetwork,86
Bing,DataBase,74
Bing,DataStructure,28
Bing,OperatingSystem,100
Bing,CLanguage,18
Bing,Python,56
Bing,ComputerNetwork,100
Bishop,Algorithm,12
Bishop,OperatingSystem,60
Blair,CLanguage,98
Blair,Python,4
Blair,ComputerNetwork,18
```

```
Blair,Software,90
Blake,DataBase,88
Blake,CLanguage,18
Blake,Python,52
Blake,ComputerNetwork,94
Blithe,DataStructure,64
Blithe,ComputerNetwork,94
Blithe,Software,86
Bob,DataBase,64
Bob,Algorithm,20
Bob,CLanguage,56
Booth,Algorithm,76
Booth,OperatingSystem,70
Booth,CLanguage,48
Booth,Python,26
Booth,ComputerNetwork,22
Booth,Software,82
Borg,DataBase,52
Borg,CLanguage,30
Borg,Python,60
Borg,ComputerNetwork,38
Boris,Algorithm,60
Boris,DataStructure,16
Boris,OperatingSystem,16
Boris,CLanguage,72
Boris,Python,10
Boris,Software,94
Bowen,DataBase,68
Bowen,Algorithm,40
Bowen,DataStructure,62
Bowen,CLanguage,26
Bowen,Python,60
Boyce,DataBase,74
Boyce,Software,6
Boyd,DataStructure,18
Boyd,OperatingSystem,94
Boyd,Software,40
Bradley,DataBase,34
Bradley,Algorithm,14
Brady,DataBase,10
Brady,Algorithm,92
Brady,DataStructure,72
Brady,CLanguage,50
Brady,Python,100
Brandon,DataBase,68
Brandon,Algorithm,74
Brandon,DataStructure,20
Brandon,OperatingSystem,80
Brandon,Software,80
Brian,Algorithm,56
Brian,DataStructure,34
Brian,OperatingSystem,12
Brian,CLanguage,2
Brian,Python,14
Brian,Software,8
Broderick,Algorithm,34
```

```
Broderick,DataStructure,32
Broderick,ComputerNetwork,48
Brook,DataStructure,72
Brook,OperatingSystem,58
Brook,CLanguage,66
Brook,Software,56
Bruce,Algorithm,100
Bruce,OperatingSystem,62
Bruce,CLanguage,26
Bruno,DataBase,98
Bruno,DataStructure,6
Bruno,CLanguage,92
Bruno,Python,68
Bruno,Software,78
Chad,DataBase,36
Chad,Algorithm,26
Chad,DataStructure,18
Chad,OperatingSystem,68
Chad,Python,36
Chad,ComputerNetwork,30
Channing,DataStructure,38
Channing,CLanguage,2
Channing,ComputerNetwork,18
Channing,Software,90
Chapman,DataBase,42
Chapman,Algorithm,42
Chapman,OperatingSystem,72
Chapman,Python,86
Charles,DataBase,36
Charles,Algorithm,14
Charles,OperatingSystem,86
Chester,DataBase,78
Chester,Algorithm,66
Chester,DataStructure,40
Chester,OperatingSystem,10
Chester,ComputerNetwork,52
Chester,Software,58
Christ,DataStructure,98
Christ,CLanguage,58
Christian,DataStructure,38
Christian,CLanguage,62
Christopher,DataBase,4
Christopher,Algorithm,22
Christopher,DataStructure,58
Christopher,Software,36
Clare,DataStructure,74
Clare,OperatingSystem,30
Clare,CLanguage,76
Clare,Software,36
Clarence,DataBase,82
Clarence,Algorithm,64
Clarence,DataStructure,98
Clarence,OperatingSystem,78
Clarence,CLanguage,22
Clarence,ComputerNetwork,92
Clarence,Software,56
```

```
Clark,DataBase,26
Clark,Algorithm,60
Clark,DataStructure,14
Clark,OperatingSystem,56
Clark,CLanguage,8
Clark,Software,44
Claude,CLanguage,52
Claude,ComputerNetwork,70
Clement,DataBase,92
Clement,OperatingSystem,8
Clement,CLanguage,86
Clement,Python,92
Clement,ComputerNetwork,16
Cleveland,DataBase,78
Cleveland,Algorithm,70
Cleveland,OperatingSystem,74
Cleveland,CLanguage,70
Cliff,Algorithm,46
Cliff,DataStructure,10
Cliff,CLanguage,52
Cliff,ComputerNetwork,74
Cliff,Software,10
Clyde,DataBase,86
Clyde,Algorithm,76
Clyde,DataStructure,82
Clyde,OperatingSystem,82
Clyde,Python,22
Clyde,ComputerNetwork,78
Clyde,Software,76
Colbert,DataBase,4
Colbert,Algorithm,4
Colbert,Python,32
Colbert,Software,12
Colby,DataBase,70
Colby,Algorithm,24
Colby,DataStructure,94
Colby,OperatingSystem,62
Colin,Algorithm,10
Colin,CLanguage,90
Colin,Python,82
Colin,ComputerNetwork,62
Colin,Software,30
Conrad,DataBase,48
Conrad,ComputerNetwork,76
Corey,DataBase,22
Corey,Algorithm,58
Corey,OperatingSystem,6
Corey,Python,94
Dean,DataBase,26
Dean,Algorithm,54
Dean,DataStructure,90
Dean,CLanguage,26
Dean,Python,98
Dean,ComputerNetwork,50
Dean,Software,82
Dempsey,DataStructure,70
```

```
Dempsey,OperatingSystem,70
Dempsey,CLanguage,98
Dempsey,ComputerNetwork,30
Dennis,Algorithm,100
Dennis,DataStructure,40
Dennis,Python,22
Dennis,ComputerNetwork,94
Derrick,DataBase,44
Derrick,Algorithm,26
Derrick,CLanguage,16
Derrick,Python,100
Derrick,ComputerNetwork,36
Derrick,Software,74
Devin,DataBase,16
Devin,DataStructure,70
Devin,Python,98
Devin,Software,0
Dick,DataStructure,62
Dick,Python,32
Dick,ComputerNetwork,2
Dominic,DataBase,16
Dominic,Python,30
Dominic,ComputerNetwork,12
Dominic,Software,24
Don,Algorithm,52
Don,ComputerNetwork,36
Donahue,DataBase,86
Donahue,DataStructure,88
Donahue,CLanguage,16
Donahue,ComputerNetwork,24
Donahue,Software,40
Donald,Algorithm,28
Donald,CLanguage,18
Donald,Python,52
Donald,ComputerNetwork,62
Drew,Algorithm,78
Drew,DataStructure,0
Drew,OperatingSystem,14
Drew,Python,28
Drew,Software,46
Duke,DataBase,14
Duke,Algorithm,28
Duke,OperatingSystem,68
Duke,CLanguage,78
Duncann,Algorithm,34
Duncann,DataStructure,86
Duncann,Python,94
Duncann,ComputerNetwork,24
Duncann,Software,78
Edward,DataBase,18
Edward,Algorithm,22
Edward,DataStructure,2
Edward,CLanguage,4
Egbert,Algorithm,26
Egbert,CLanguage,24
Egbert,Python,92
```

```
Egbert,ComputerNetwork,12
Eli,DataBase,54
Eli,Algorithm,54
Eli,CLanguage,94
Eli,Python,60
Eli,ComputerNetwork,30
Elijah,CLanguage,30
Elijah,Python,62
Elijah,ComputerNetwork,96
Elijah,Software,36
Elliot,Algorithm,60
Elliot,OperatingSystem,96
Elliot,Software,78
Ellis,Algorithm,90
Ellis,OperatingSystem,36
Ellis,ComputerNetwork,56
Ellis,Software,28
Elmer,DataStructure,34
Elmer,CLanguage,98
Elmer,Python,22
Elmer,ComputerNetwork,44
Elroy,DataBase,48
Elroy,Algorithm,82
Elroy,DataStructure,44
Elroy,OperatingSystem,56
Elroy,CLanguage,78
Elton,DataBase,80
Elton,DataStructure,2
Elton,OperatingSystem,16
Elton,CLanguage,44
Elton,Python,40
Elvis,DataBase,32
Elvis,DataStructure,20
Emmanuel,DataBase,32
Emmanuel,OperatingSystem,42
Emmanuel,CLanguage,12
Enoch,DataBase,54
Enoch,Algorithm,22
Enoch,Python,78
Eric,DataBase,18
Eric,Algorithm,62
Eric,ComputerNetwork,68
Eric,Software,64
Ernest,DataBase,62
Ernest,OperatingSystem,6
Ernest,CLanguage,70
Ernest,Python,94
Ernest,ComputerNetwork,16
Eugene,CLanguage,80
Evan,DataStructure,8
Evan,OperatingSystem,100
Evan,Python,20
Ford,DataBase,32
Ford,Algorithm,66
Ford,Python,68
Francis,DataBase,58
```

```
Francis,OperatingSystem,78
Francis,CLanguage,6
Francis,Software,76
Frank,DataBase,74
Frank,Python,58
Frank,ComputerNetwork,60
Geoffrey,OperatingSystem,4
Geoffrey,CLanguage,24
Geoffrey,Python,86
Geoffrey,Software,52
George,Algorithm,72
George,DataStructure,80
George,Python,36
George,ComputerNetwork,50
Gerald,Algorithm,46
Gerald,OperatingSystem,94
Gerald,CLanguage,90
Gerald,ComputerNetwork,8
Gilbert,Algorithm,80
Gilbert,CLanguage,96
Gilbert,ComputerNetwork,72
Giles,DataBase,6
Giles,Algorithm,12
Giles,DataStructure,26
Giles,CLanguage,6
Giles,Python,72
Giles,ComputerNetwork,18
Giles,Software,78
Glenn,DataBase,12
Glenn,Algorithm,42
Glenn,OperatingSystem,82
Glenn,CLanguage,20
Glenn,Python,84
Glenn,ComputerNetwork,76
Gordon,DataBase,60
Gordon,Algorithm,64
Gordon,OperatingSystem,38
Gordon,Python,48
Greg,Algorithm,18
Greg,DataStructure,28
Greg,Python,78
Greg,Software,72
Griffith,Algorithm,40
Griffith,DataStructure,58
Griffith,OperatingSystem,10
Griffith,Software,4
Harlan,Algorithm,44
Harlan,OperatingSystem,46
Harlan,CLanguage,86
Harlan,Python,86
Harlan,ComputerNetwork,56
Harlan,Software,12
Harold,DataStructure,78
Harold,OperatingSystem,100
Harold,CLanguage,52
Harold,Python,12
```

```
Harry,DataBase,74
Harry,OperatingSystem,60
Harry,Python,42
Harry,Software,46
Harvey,DataBase,86
Harvey,Algorithm,88
Harvey,DataStructure,40
Harvey,OperatingSystem,74
Harvey,Python,14
Harvey,ComputerNetwork,78
Harvey,Software,22
Hayden,Algorithm,36
Hayden,DataStructure,80
Hayden,Software,34
Henry,Python,4
Henry,ComputerNetwork,74
Herbert,OperatingSystem,88
Herbert,CLanguage,26
Herbert,ComputerNetwork,18
Herman,OperatingSystem,24
Herman,ComputerNetwork,14
Herman,Software,78
Hilary,DataStructure,58
Hilary,Python,2
Hilary,ComputerNetwork,98
Hilary,Software,32
Hiram,DataBase,12
Hiram,Algorithm,44
Hiram,DataStructure,74
Hiram,OperatingSystem,70
Hiram,CLanguage,46
Hiram,ComputerNetwork,38
Hobart,DataBase,26
Hobart,Algorithm,0
Hobart,DataStructure,44
Hobart,ComputerNetwork,48
Hogan,DataBase,80
Hogan,CLanguage,40
Hogan,Python,10
Hogan,Software,26
Horace,DataBase,22
Horace,OperatingSystem,52
Horace,CLanguage,54
Horace,ComputerNetwork,10
Horace,Software,24
Ivan,OperatingSystem,70
Ivan,Python,10
Ivan,ComputerNetwork,100
Ivan,Software,36
Jason,Algorithm,38
Jason,OperatingSystem,18
Jason,CLanguage,8
Jason,ComputerNetwork,4
Jay,Algorithm,58
Jay,DataStructure,30
Jay,OperatingSystem,24
```

```
Jay,CLanguage,22
Jay,Python,38
Jay,Software,6
Jeff,DataBase,20
Jeff,DataStructure,0
Jeff,ComputerNetwork,18
Jeff,Software,16
Jeffrey,DataStructure,66
Jeffrey,OperatingSystem,4
Jeffrey,CLanguage,100
Jeffrey,Software,86
Jeremy,DataBase,84
Jeremy,Algorithm,44
Jeremy,DataStructure,90
Jeremy,CLanguage,94
Jeremy,Python,60
Jeremy,Software,66
Jerome,DataBase,16
Jerome,DataStructure,64
Jerome,OperatingSystem,10
Jerry,DataStructure,30
Jerry,Python,46
Jerry,ComputerNetwork,94
Jesse,Algorithm,78
Jesse,DataStructure,50
Jesse,OperatingSystem,14
Jesse,CLanguage,100
Jesse,Python,28
Jesse,ComputerNetwork,94
Jesse,Software,84
Jim,Algorithm,32
Jim,OperatingSystem,36
Jim,Python,4
Jim,ComputerNetwork,38
Jo,DataBase,14
Jo,DataStructure,52
Jo,OperatingSystem,68
Jo,CLanguage,92
Jo,ComputerNetwork,28
John,DataBase,60
John,Algorithm,14
John,OperatingSystem,64
John,Python,34
John,ComputerNetwork,34
John,Software,36
Jonas,Algorithm,38
Jonas,Python,84
Jonas,ComputerNetwork,0
Jonas,Software,44
Jonathan,OperatingSystem,74
Jonathan,CLanguage,38
Jonathan,Python,86
Jonathan,Software,30
Joseph,DataStructure,30
Joseph,CLanguage,28
Joseph,ComputerNetwork,84
```

```
Joshua,Algorithm,30
Joshua,DataStructure,46
Joshua,OperatingSystem,74
Joshua,Software,0
Ken,Algorithm,74
Ken,OperatingSystem,60
Ken,CLanguage,68
Kennedy,DataBase,68
Kennedy,DataStructure,32
Kennedy,OperatingSystem,20
Kennedy,Python,14
Kenneth,OperatingSystem,74
Kenneth,CLanguage,18
Kenneth,ComputerNetwork,34
Kent,DataBase,82
Kent,DataStructure,50
Kent,CLanguage,34
Kent,Python,20
Kerr,Algorithm,70
Kerr,Python,32
Kerr,ComputerNetwork,36
Kerr,Software,36
Kerwin,Algorithm,64
Kerwin,OperatingSystem,24
Kerwin,ComputerNetwork,58
Kevin,DataBase,54
Kevin,DataStructure,44
Kevin,CLanguage,6
Kevin,Software,26
Kim,DataBase,0
Kim,Algorithm,40
Kim,DataStructure,14
Kim,Python,6
Len,DataBase,60
Len,OperatingSystem,22
Len,Python,88
Len,ComputerNetwork,76
Len,Software,92
Lennon,DataBase,84
Lennon,Algorithm,2
Lennon,OperatingSystem,98
Lennon,Software,42
Leo,DataBase,44
Leo,OperatingSystem,42
Leo,CLanguage,46
Leo,Python,38
Leo,Software,20
Leonard,Algorithm,96
Leonard,Software,20
Leopold,DataBase,48
Leopold,Algorithm,38
Leopold,DataStructure,96
Leopold,CLanguage,24
Leopold,Python,52
Leopold,ComputerNetwork,90
Leopold,Software,94
```

```
Les,DataBase,72
Les,Algorithm,58
Les,DataStructure,26
Les,CLanguage,2
Les,Python,38
Les,ComputerNetwork,20
Lester,DataStructure,100
Lester,CLanguage,100
Lester,Python,96
Lester,ComputerNetwork,50
Levi,CLanguage,36
Levi,Software,86
Lewis,Algorithm,62
Lewis,DataStructure,60
Lewis,OperatingSystem,18
Lewis,Python,60
Lionel,DataStructure,82
Lionel,OperatingSystem,88
Lionel,CLanguage,22
Lionel,ComputerNetwork,22
Lou,OperatingSystem,88
Lou,Software,52
Louis,DataBase,50
Louis,Algorithm,76
Louis,DataStructure,32
Louis,OperatingSystem,18
Louis,Python,56
Louis,Software,94
Lucien,DataStructure,22
Lucien,CLanguage,58
Lucien,Python,94
Lucien,ComputerNetwork,94
Lucien,Software,58
Luthers,Algorithm,44
Luthers,DataStructure,16
Luthers,OperatingSystem,84
Luthers,CLanguage,22
Luthers,ComputerNetwork,88
Marico,DataBase,56
Marico,Algorithm,56
Marico,DataStructure,16
Marico,CLanguage,40
Marico,ComputerNetwork,18
Marico,Software,24
Mark,DataBase,66
Mark,Algorithm,46
Mark,DataStructure,36
Mark,OperatingSystem,86
Mark,Python,84
Mark,ComputerNetwork,30
Mark,Software,60
Marlon,DataStructure,44
Marlon,OperatingSystem,52
Marlon,CLanguage,34
Marlon,Software,62
Marsh,Algorithm,64
```

```
Marsh,Python,86
Marsh,ComputerNetwork,68
Marsh,Software,42
Marshall,DataBase,38
Marshall,OperatingSystem,38
Marshall,CLanguage,50
Marshall,Software,76
Martin,CLanguage,84
Martin,Python,98
Martin,Software,38
Marvin,Algorithm,12
Marvin,OperatingSystem,82
Marvin,CLanguage,64
Matt,DataBase,46
Matt,DataStructure,48
Matt,CLanguage,22
Matt,Python,100
Matthew,CLanguage,14
Matthew,ComputerNetwork,48
Maurice,DataStructure,26
Maurice,ComputerNetwork,16
Max,Algorithm,32
Max,DataStructure,38
Max,ComputerNetwork,36
Maxwell,OperatingSystem,78
Maxwell,Python,52
Maxwell,ComputerNetwork,82
Maxwell,Software,22
Meredith,DataBase,26
Meredith,Algorithm,42
Meredith,OperatingSystem,42
Meredith,Python,52
Merle,OperatingSystem,12
Merle,ComputerNetwork,40
Merle,Software,4
Merlin,Algorithm,62
Merlin,DataStructure,2
Merlin,OperatingSystem,90
Merlin,ComputerNetwork,60
Merlin,Software,20
Michael,Algorithm,92
Michael,CLanguage,66
Michael,Python,6
Michael,ComputerNetwork,42
Michael,Software,98
Mick,DataStructure,64
Mick,OperatingSystem,98
Mick,Python,2
Mick,Software,76
Mike,Algorithm,92
Mike,DataStructure,56
Mike,ComputerNetwork,62
Miles,DataBase,56
Miles,Algorithm,76
Miles,DataStructure,66
Miles,OperatingSystem,60
```

```
Miles,Python,32
Miles,ComputerNetwork,80
Milo,CLanguage,68
Milo,Python,64
Monroe,DataBase,42
Monroe,Algorithm,16
Monroe,ComputerNetwork,28
Montague,Algorithm,36
Montague,OperatingSystem,24
Montague,ComputerNetwork,16
Nelson,DataBase,40
Nelson,Algorithm,80
Nelson,DataStructure,16
Nelson,OperatingSystem,24
Nelson,Python,36
Newman,Algorithm,84
Newman,Software,52
Nicholas,DataBase,24
Nicholas,Algorithm,38
Nicholas,DataStructure,58
Nicholas,OperatingSystem,78
Nicholas,CLanguage,100
Nick,OperatingSystem,100
Nick,CLanguage,56
Nick,Python,12
Nick,ComputerNetwork,92
Nick,Software,64
Nigel,Algorithm,4
Nigel,ComputerNetwork,10
Nigel,Software,4
Noah,DataBase,80
Noah,OperatingSystem,54
Noah,CLanguage,44
Noah,Python,22
Payne,DataBase,50
Payne,Algorithm,30
Payne,DataStructure,62
Payne,Python,94
Payne,ComputerNetwork,92
Payne,Software,80
Perry,DataStructure,38
Perry,OperatingSystem,88
Perry,CLanguage,18
Perry,ComputerNetwork,68
Perry,Software,98
Pete,DataStructure,10
Pete,OperatingSystem,42
Pete,Software,74
Peter,DataBase,88
Peter,Algorithm,46
Peter,DataStructure,58
Peter,Software,54
Phil,DataBase,16
Phil,OperatingSystem,16
Phil,Software,14
Philip,DataBase,24
```

```
Philip,OperatingSystem,30
Randolph,Algorithm,18
Randolph,DataStructure,82
Randolph,OperatingSystem,90
Raymondt,DataBase,86
Raymondt,Algorithm,54
Raymondt,DataStructure,78
Raymondt,CLanguage,46
Raymondt,Python,78
Raymondt,Software,100
Robin,Algorithm,68
Robin,DataStructure,2
Robin,Python,90
Robin,Software,54
Rock,DataBase,6
Rock,Algorithm,92
Rock,OperatingSystem,88
Rock,CLanguage,0
Rock,Python,94
Rock,Software,98
Rod,Algorithm,84
Rod,OperatingSystem,94
Rod,Python,18
Rod,ComputerNetwork,56
Roderick,DataBase,50
Roderick,Algorithm,62
Roderick,OperatingSystem,66
Roderick,CLanguage,12
Rodney,Algorithm,34
Rodney,OperatingSystem,52
Rodney,ComputerNetwork,44
Ron,DataBase,82
Ron,Algorithm,76
Ron,DataStructure,36
Ron,CLanguage,58
Ron,Python,40
Ron,ComputerNetwork,36
Ronald,DataBase,66
Ronald,Algorithm,20
Ronald,CLanguage,32
Rory,Algorithm,68
Rory,OperatingSystem,12
Rory,CLanguage,90
Rory,Software,76
Roy,DataBase,88
Roy,DataStructure,58
Roy,OperatingSystem,20
Roy,CLanguage,74
Roy,Python,70
Roy,ComputerNetwork,0
Samuel,DataBase,66
Samuel,Algorithm,32
Samuel,OperatingSystem,20
Samuel,ComputerNetwork,96
Sandy,DataStructure,72
Saxon,DataBase,44
```

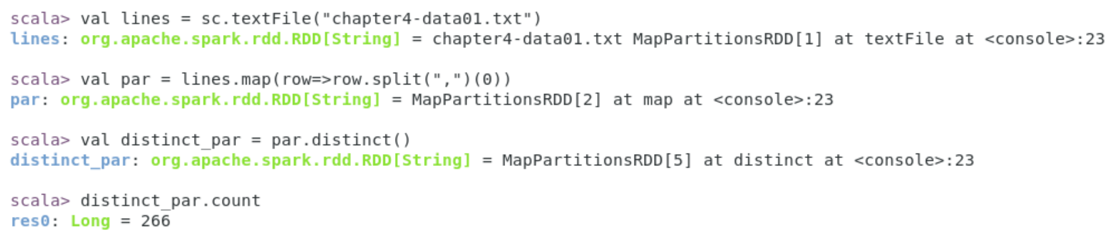
```
Saxon,Algorithm,52
Saxon,DataStructure,52
Saxon,OperatingSystem,46
Saxon,CLanguage,60
Saxon,ComputerNetwork,66
Saxon,Software,38
Scott,Algorithm,46
Scott,OperatingSystem,78
Scott,Software,4
Sean,DataBase,62
Sean,Algorithm,92
Sean,OperatingSystem,92
Sean,CLanguage,0
Sean,Python,62
Sean,ComputerNetwork,34
Sebastian,DataBase,68
Sebastian,Algorithm,38
Sebastian,OperatingSystem,62
Sebastian,CLanguage,10
Sebastian,Python,64
Sebastian,ComputerNetwork,100
Sid,DataBase,14
Sid,OperatingSystem,20
Sid,CLanguage,88
Sidney,DataBase,96
Sidney,Algorithm,36
Sidney,DataStructure,8
Sidney,ComputerNetwork,0
Sidney,Software,34
Simon,ComputerNetwork,96
Simon,Software,64
Solomon,DataBase,2
Solomon,Algorithm,46
Solomon,DataStructure,20
Solomon,ComputerNetwork,64
Solomon,Software,18
Spencer,DataStructure,24
Spencer,OperatingSystem,88
Spencer,CLanguage,96
Spencer,Python,14
Spencer,ComputerNetwork,98
Stan,DataStructure,64
Stan,CLanguage,48
Stan,Python,46
Todd,OperatingSystem,82
Todd,Python,52
Todd,ComputerNetwork,42
Tom,DataBase,26
Tom,Algorithm,12
Tom,OperatingSystem,16
Tom,Python,40
Tom,Software,60
Tony,DataBase,30
Tony,Algorithm,12
Tony,Python,96
Tracy,DataBase,34
```

```
Tracy,CLanguage,72
Tracy,Software,74
Truman,Algorithm,60
Truman,Python,74
Truman,ComputerNetwork,54
Upton,DataBase,94
Upton,Algorithm,52
Upton,DataStructure,28
Upton,Python,86
Upton,ComputerNetwork,78
Uriah,Algorithm,54
Valentine,DataBase,10
Valentine,DataStructure,76
Valentine,CLanguage,96
Valentine,Python,38
Valentine,Software,60
Valentine,DataBase,0
Valentine,DataStructure,40
Valentine,CLanguage,56
Verne,OperatingSystem,30
Verne,Python,74
Verne,Software,94
Vic,DataBase,62
Vic,CLanguage,56
Vic,ComputerNetwork,66
Victor,ComputerNetwork,42
Victor,Software,6
Vincent,DataBase,70
Vincent,Algorithm,98
Vincent,OperatingSystem,48
Vincent,ComputerNetwork,64
Vincent,Software,48
Virgil,DataStructure,30
Virgil,OperatingSystem,8
Virgil,Python,22
Virgil,ComputerNetwork,68
Virgil,Software,60
Walter,DataBase,96
Walter,Algorithm,34
Walter,OperatingSystem,62
Walter,Software,4
Ward,DataStructure,38
Ward,OperatingSystem,64
Ward,ComputerNetwork,96
Ward,Software,88
Webb,DataBase,26
Webb,Algorithm,32
Webb,DataStructure,94
Webb,CLanguage,38
Webb,Python,44
Webb,ComputerNetwork,42
Webb,Software,84
Webster,OperatingSystem,98
Webster,Software,16
Will,Algorithm,30
Will,OperatingSystem,96
```

```
Will,CLanguage,38
William,DataBase,74
William,DataStructure,36
William,OperatingSystem,58
William,CLanguage,98
William,ComputerNetwork,68
William,Software,74
Willie,DataStructure,24
Willie,OperatingSystem,70
Willie,Python,48
Willie,ComputerNetwork,92
Winfred,Algorithm,16
Winfred,CLanguage,22
Winfred,Software,26
Winston,DataStructure,66
Winston,OperatingSystem,26
Winston,CLanguage,98
Winston,Software,40
Woodrow,DataBase,26
Woodrow,OperatingSystem,72
Woodrow,Python,44
Wordsworth,DataStructure,50
Wordsworth,OperatingSystem,62
Wordsworth,Python,42
Wordsworth,ComputerNetwork,4
Wright,DataBase,76
Wright,OperatingSystem,100
Wright,ComputerNetwork,44
Wright,Software,60
```

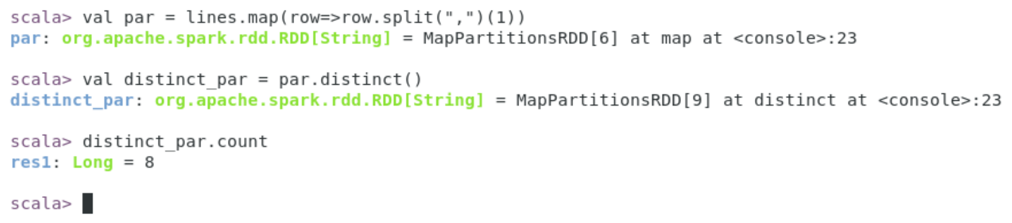
请根据给定的实验数据，在`spark-shell`中通过编程来计算以下内容：

1. 该系总共有多少学生；



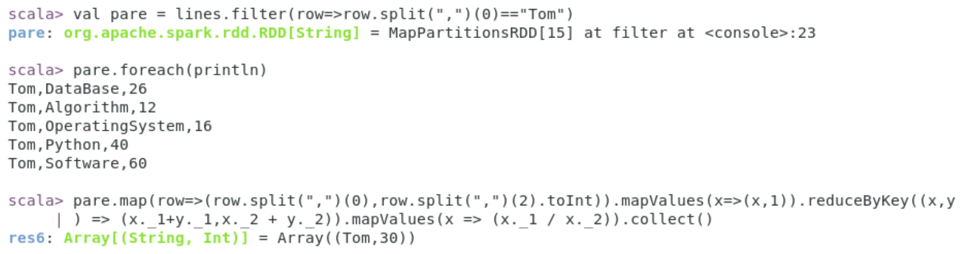
```
scala> val lines = sc.textFile("chapter4-data01.txt")
lines: org.apache.spark.rdd.RDD[String] = chapter4-data01.txt MapPartitionsRDD[1] at textFile at <console>:23

scala> val par = lines.map(row=>row.split(",")(0))
par: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at map at <console>:23

scala> val distinct_par = par.distinct()
distinct_par: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[5] at distinct at <console>:23

scala> distinct_par.count
res0: Long = 266
```

2. 该系共开设来多少门课程；



```
scala> val par = lines.map(row=>row.split(",")(1))
par: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[6] at map at <console>:23

scala> val distinct_par = par.distinct()
distinct_par: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[9] at distinct at <console>:23

scala> distinct_par.count
res1: Long = 8

scala> 
```

3. Tom同学的总成绩平均分是多少；

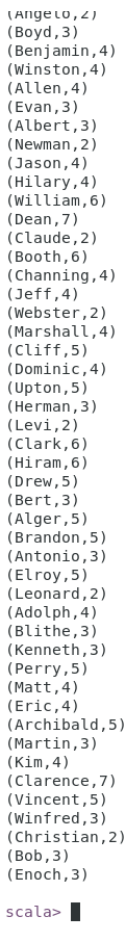


```
scala> val pare = lines.filter(row=>row.split(",")(0)=="Tom")
pare: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[15] at filter at <console>:23

scala> pare.foreach(println)
Tom,DataBase,26
Tom,Algorithm,12
Tom,OperatingSystem,16
Tom,Python,40
Tom,Software,60

scala> pare.map(row=>(row.split(",")(0),row.split(",")(2).toInt)).mapValues(x=>(x,1)).reduceByKey((x,y
     | ) => (x._1+y._1,x._2 + y._2)).mapValues(x => (x._1 / x._2)).collect()
res6: Array[(String, Int)] = Array((Tom,30))
```

4. 求每名同学的选修的课程门数;

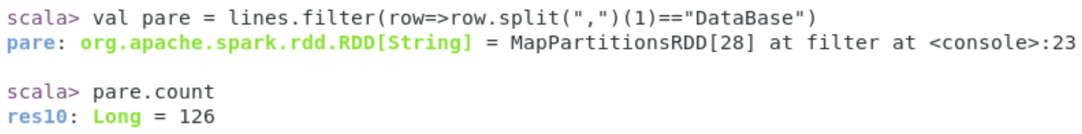
```
scala> val pare = lines.map(row=>(row.split(",")(0),row.split(",")(1)))
pare: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[24] at map at <console>:23

scala> pare.mapValues(x => (x,1)).reduceByKey((x,y) => (" ",x._2 + y._2)).mapValues(x =>
     | x._2).foreach(println)
(Ford,3)
(Lionel,4)
(Verne,3)
(Lennon,4)
(Joshua,4)
(Marvin,3)
(Marsh,4)
(Bartholomew,5)
(Conrad,2)
(Armand,3)
(Jonathan,4)
(Broderick,3)
(Brady,5)
(Derrick,6)
(Rod,4)
(Willie,4)
(Walter,4)
(Boyce,2)
(Duncann,5)
(Elvis,2)
(Elmer,4)
(Bennett,6)
(Elton,5)
(Jo,5)
(Jim,4)
(Adonis,5)
(Abel,4)
(Peter,4)
(Alvis,6)
(Joseph,3)
(Raymondt,6)
(Kerwin,3)
(Wright,4)
(Adam,3)
(Borg,4)
(Sandy,1)
(Ben,4)
(Miles,6)
(Clyde,7)
(Francis,4)
(Dempsey,4)
(Ellis,4)
(Edward,4)
(Mick,4)
(Cleveland,4)
(Luthers,5)
(Virgil,5)
(Ivan,4)
(Alvin,5)
(Dick,3)
(Bevis,4)
(Leo,5)
(Saxon,7)
(Armstrong,2)
(Hogan,4)
(Sid,3)
(Blair,4)
(Colbert,4)
(Lucien,5)
(Kerr,4)
(Montague,3)
(Giles,7)
(Kevin,4)
(Uriah,1)
(Jeffrey,4)
(Simon,2)
(Elijah,4)
(Greg,4)
(Colin,5)
(Arlen,4)
(Maxwell,4)
(Payne,6)
(Kennedy,4)
(Spencer,5)
(Kent,4)
(Griffith,4)
(Jeremy,6)
(Alan,5)
(Andrew,4)
(Jerry,3)
(Donahue,5)
(Gilbert,3)
(Bishop,2)
(Bernard,2)
(Egbert,4)
(George,4)
(Noah,4)
(Bruce,3)
(Mike,3)
(Frank,3)
(Boris,6)
(Tony,3)
(Christ,2)
(Ken,3)
(Milo,2)
(Victor,2)
(Clare,4)
(Nigel,3)
(Christopher,4)
(Robin,4)
(Chad,6)
(Alfred,2)
(Woodrow,3)
(Rory,4)
(Dennis,4)
(Ward,4)
(Chester,6)
```

```
(Emmanuel,3)
(Stan,3)
(Jerome,3)
(Corey,4)
(Harvey,7)
(Herbert,3)
(Maurice,2)
(Merle,3)
(Les,6)
(Bing,6)
(Charles,3)
(Clement,5)
(Leopold,7)
(Brian,6)
(Horace,5)
(Sebastian,6)
(Bernie,3)
(Basil,4)
(Michael,5)
(Ernest,5)
(Tom,5)
(Vic,3)
(Eli,5)
(Duke,4)
(Alva,5)
(Lester,4)
(Hayden,3)
(Bertram,3)
(Bart,5)
(aron,1)
(Sidney,5)
(Bowen,5)
(Roderick,4)
(Colby,4)
(Jay,6)
(Meredith,4)
(Harold,4)
(Max,3)
(Adair,3)
(Scott,3)
(Barton,1)
(Elliot,3)
(Matthew,2)
(Alexander,4)
(Todd,3)
(Wordsworth,4)
(Geoffrey,4)
(Devin,4)
(Donald,4)
(Roy,6)
(Harry,4)
(Abbott,3)
(Baron,6)
(Mark,7)
(Lewis,4)
(Rock,6)
(Eugene,1)
(Aries,2)
(Samuel,4)
(Glenn,6)
(Will,3)
(Gerald,4)
(Henry,2)
(Jesse,7)
(Bradley,2)
(Merlin,5)
(Monroe,3)
(Hobart,4)
(Ron,6)
(Archer,5)
(Nick,5)
(Louis,6)
(Len,5)
(Randolph,3)
(Benson,4)
(John,6)
(Abraham,3)
(Benedict,6)
(Marico,6)
(Berg,4)
(Aldrich,3)
(Lou,2)
(Brook,4)
(Ronald,3)
(Pete,3)
(Nicholas,5)
(Bill,2)
(Harlan,6)
(Tracy,3)
(Gordon,4)
(Alston,4)
(Andy,3)
(Bruno,5)
(Beck,4)
(Phil,3)
(Barry,5)
(Nelson,5)
(Antony,5)
(Rodney,3)
(Truman,3)
(Marlon,4)
(Don,2)
(Philip,2)
(Sean,6)
(Webb,7)
(Solomon,5)
(Aaron,3)
(Blake,4)
(Amos,5)
(Chapman,4)
(Jonas,4)
(Valentine,8)
```



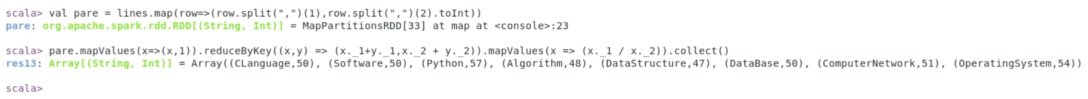
```
(Angelo,2)
(Boyd,3)
(Benjamin,4)
(Winston,4)
(Allen,4)
(Evan,3)
(Albert,3)
(Newman,2)
(Jason,4)
(Hilary,4)
(William,6)
(Dean,7)
(Claude,2)
(Booth,6)
(Channing,4)
(Jeff,4)
(Webster,2)
(Marshall,4)
(Cliff,5)
(Dominic,4)
(Upton,5)
(Herman,3)
(Levi,2)
(Clark,6)
(Hiram,6)
(Drew,5)
(Bert,3)
(Alger,5)
(Brandon,5)
(Antonio,3)
(Elroy,5)
(Leonard,2)
(Adolph,4)
(Blithe,3)
(Kenneth,3)
(Perry,5)
(Matt,4)
(Eric,4)
(Archibald,5)
(Martin,3)
(Kim,4)
(Clarence,7)
(Vincent,5)
(Winfred,3)
(Christian,2)
(Bob,3)
(Enoch,3)

scala>
```

5. 该系`DataBase`课程共有多少人选修；



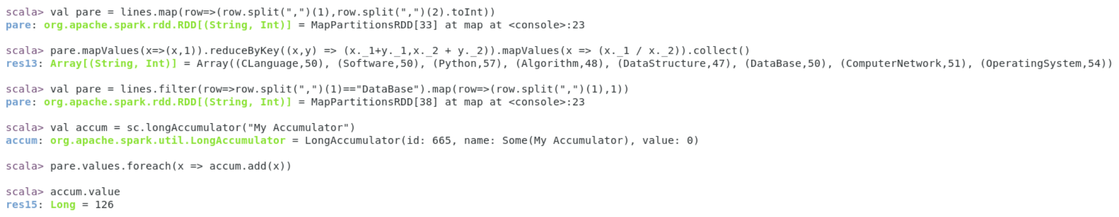
```
scala> val pare = lines.filter(row=>row.split(",")(1)=="DataBase")
pare: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[28] at filter at <console>:23

scala> pare.count
res10: Long = 126
```

6. 各门课程的平均分是多少；



```
scala> val pare = lines.map(row=>(row.split(",")(1),row.split(",")(2).toInt))
pare: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[33] at map at <console>:23

scala> pare.mapValues(x=>(x,1)).reduceByKey((x,y) => (x._1+y._1,x._2 + y._2)).mapValues(x => (x._1 / x._2)).collect()
res13: Array[(String, Int)] = Array((CLanguage,50), (Software,50), (Python,57), (Algorithm,48), (DataStructure,47), (DataBase,50), (ComputerNetwork,51), (OperatingSystem,54))

scala>
```

7. 使用累加器计算共有多少人选了`DataBase`这门课。



```
scala> val pare = lines.map(row=>(row.split(",")(1),row.split(",")(2).toInt))
pare: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[33] at map at <console>:23

scala> pare.mapValues(x=>(x,1)).reduceByKey((x,y) => (x._1+y._1,x._2 + y._2)).mapValues(x => (x._1 / x._2)).collect()
res13: Array[(String, Int)] = Array((CLanguage,50), (Software,50), (Python,57), (Algorithm,48), (DataStructure,47), (DataBase,50), (ComputerNetwork,51), (OperatingSystem,54))

scala> val pare = lines.filter(row=>row.split(",")(1)=="DataBase").map(row=>(row.split(",")(1),1))
pare: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[38] at map at <console>:23

scala> val accum = sc.longAccumulator("My Accumulator")
accum: org.apache.spark.util.LongAccumulator = LongAccumulator(id: 665, name: Some(My Accumulator), value: 0)

scala> pare.values.foreach(x => accum.add(x))

scala> accum.value
res15: Long = 126
```

2. 编写独立应用程序实现数据去重

对于两个输入文件`A`和`B`，编写`Spark`独立应用程序，对两个文件进行合并，并剔除其中重复的内容，得到一个新文件`C`。下面是输入文件和输出文件的一个样例，供参考。

输入文件`A`的样例如下：

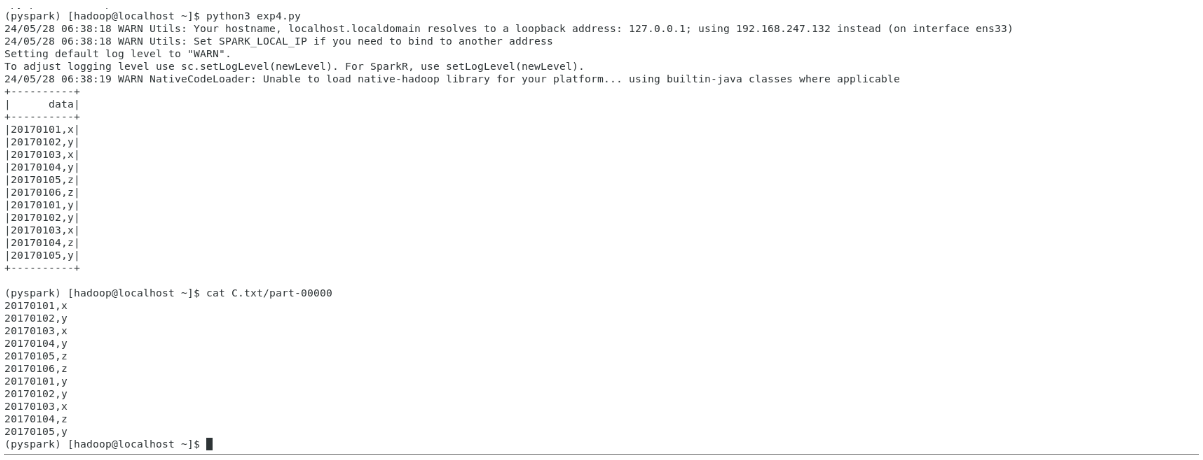
```
20170101 x
20170102 y
20170103 x
20170104 y
20170105 z
20170106 z
```

输入文件`B`的样例如下：

```
20170101 y
20170102 y
20170103 x
20170104 z
20170105 y
```

根据输入的文件`A`和`B`合并得到的输出文件`C`的样例如下：

```
20170101 x
20170101 y
20170102 y
20170103 x
20170104 y
20170104 z
20170105 y
20170105 z
20170106 z
```



```
(pyspark) [hadoop@localhost ~]$ python3 exp4.py
24/05/28 06:38:18 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.247.132 instead (on interface ens33)
24/05/28 06:38:18 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/05/28 06:38:19 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
+----------+
|      data|
+----------+
|20170101,x|
|20170102,y|
|20170103,x|
|20170104,y|
|20170105,z|
|20170106,z|
|20170101,y|
|20170102,y|
|20170103,x|
|20170104,z|
|20170105,y|
+----------+

(pyspark) [hadoop@localhost ~]$ cat C.txt/part-00000
20170101,x
20170102,y
20170103,x
20170104,y
20170105,z
20170106,z
20170101,y
20170102,y
20170103,x
20170104,z
20170105,y
(pyspark) [hadoop@localhost ~]$ 
```

`python`代码如下：

```python
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, StringType
from pyspark.sql.functions import concat, lit
# 创建SparkSession
spark = SparkSession.builder.appName("MergeFiles").getOrCreate()

# 定义schema
schema = StructType([
    StructField("date", StringType(), True),
    StructField("value", StringType(), True)
])
```

```
# 读取数据并创建RDD
rddA = spark.sparkContext.textFile("/home/hadoop/A.txt") \
    .map(lambda line: line.split(" ")).map(lambda x: (x[0], x[1]))

rddB = spark.sparkContext.textFile("/home/hadoop/B.txt") \
    .map(lambda line: line.split(" ")).map(lambda x: (x[0], x[1]))

# 合并两个RDD
mergedRDD = rddA.union(rddB)

# 将合并后的RDD转换为DataFrame
df = spark.createDataFrame(mergedRDD, schema)
df = df.withColumn("data", concat(df["date"], lit(","),
df["value"])).select("data")
df.show()
df.rdd.map(lambda x: x[0]).coalesce(1, True).saveAsTextFile("/home/hadoop/C.txt")

# 关闭SparkSession
spark.stop()
```

3. 编写独立应用程序实现求平均值问题

每个输入文件表示班级学生某个学科的成绩，每行内容由两个字段组成，第一个是学生名字，第二个是学生的成绩；编写`Spark`独立应用程序求出所有学生的平均成绩，并输出到一个新文件中。下面是输入文件和输出文件的一个样例，供参考。

`Algorithm`成绩：

```
小明 92
小红 87
小新 82
小丽 90
```

`Database`成绩：

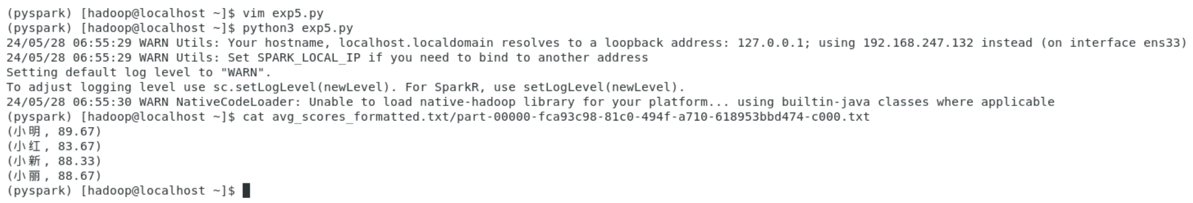
```
小明 95
小红 81
小新 89
小丽 85
```

`Python`成绩：

```
小明 82
小红 83
小新 94
小丽 91
```

平均成绩如下：

```
(小红,83.67)
(小新,88.33)
(小明,89.67)
(小丽,88.67)
```



```
(pyspark) [hadoop@localhost ~]$ vim exp5.py
(pyspark) [hadoop@localhost ~]$ python3 exp5.py
24/05/28 06:55:29 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.247.132 instead (on interface ens33)
24/05/28 06:55:29 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/05/28 06:55:30 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
(pyspark) [hadoop@localhost ~]$ cat avg_scores_formatted.txt/part-00000-fca93c98-81c0-494f-a710-618953bbd474-c000.txt
(小明, 89.67)
(小红, 83.67)
(小新, 88.33)
(小丽, 88.67)
(pyspark) [hadoop@localhost ~]$ 
```

`python`代码如下：

```python
from pyspark.sql import SparkSession
from pyspark.sql import functions as F

# 创建SparkSession
spark = SparkSession.builder.appName("AverageScore").getOrCreate()

# 读取输入文件并创建DataFrame
algorithm_df = spark.read.text("/home/hadoop/Algorithm成绩.txt")
database_df = spark.read.text("/home/hadoop/Database成绩.txt")
python_df = spark.read.text("/home/hadoop/Python成绩.txt")

# 将数据按空格分割并创建DataFrame
algorithm_split_df = algorithm_df.withColumn("name",
F.split(algorithm_df['value'], ' ')[0]).withColumn("algorithm_score",
F.split(algorithm_df['value'], ' ')[1].cast("double"))
database_split_df = database_df.withColumn("name", F.split(database_df['value'],
' ')[0]).withColumn("database_score", F.split(database_df['value'], ' ')
[1].cast("double"))
python_split_df = python_df.withColumn("name", F.split(python_df['value'], ' ')
[0]).withColumn("python_score", F.split(python_df['value'], ' ')
[1].cast("double"))

# 计算平均成绩
merged_df = algorithm_split_df.join(database_split_df,
"name").join(python_split_df, "name")
merged_df = merged_df.withColumn("avg_score", F.round((F.col("algorithm_score") +
F.col("database_score") + F.col("python_score")) / 3, 2))

# 选择结果并输出到新文件
result_df = merged_df.select(F.concat(F.lit("("), F.col("name"), F.lit(", "),
F.col("avg_score"), F.lit(")")).alias("result"))

# 将结果写入文本文件
result_df.write.mode("overwrite").text("/home/hadoop/avg_scores_formatted.txt")

# 关闭SparkSession
spark.stop()
```