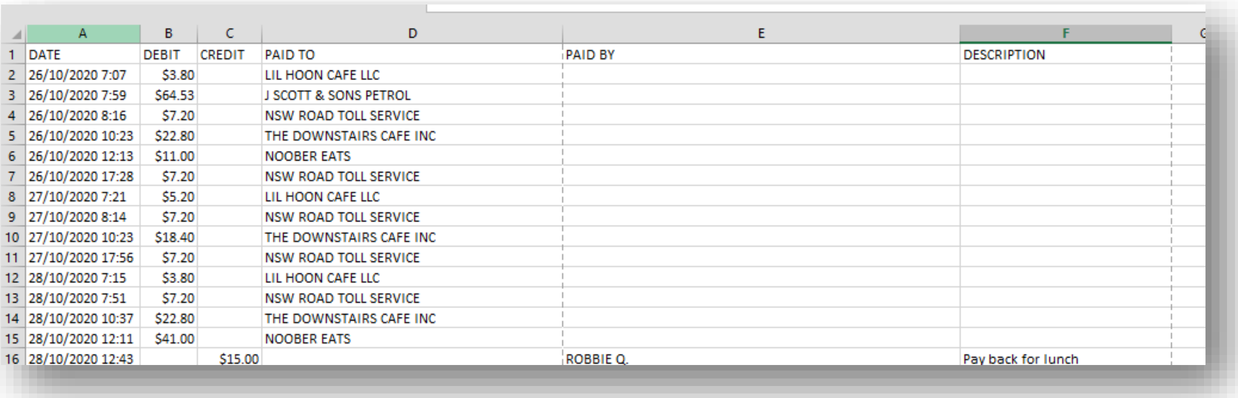
BRIEF



	A	B	C	D	E	F
1	DATE	DEBIT	CREDIT	PAID TO	PAID BY	DESCRIPTION
2	26/10/2020 7:07	$3.80		LIL HOON CAFE LLC		
3	26/10/2020 7:59	$64.53		J SCOTT & SONS PETROL		
4	26/10/2020 8:16	$7.20		NSW ROAD TOLL SERVICE		
5	26/10/2020 10:23	$22.80		THE DOWNSTAIRS CAFE INC		
6	26/10/2020 12:13	$11.00		NOOBER EATS		
7	26/10/2020 17:28	$7.20		NSW ROAD TOLL SERVICE		
8	27/10/2020 7:21	$5.20		LIL HOON CAFE LLC		
9	27/10/2020 8:14	$7.20		NSW ROAD TOLL SERVICE		
10	27/10/2020 10:23	$18.40		THE DOWNSTAIRS CAFE INC		
11	27/10/2020 17:56	$7.20		NSW ROAD TOLL SERVICE		
12	28/10/2020 7:15	$3.80		LIL HOON CAFE LLC		
13	28/10/2020 7:51	$7.20		NSW ROAD TOLL SERVICE		
14	28/10/2020 10:37	$22.80		THE DOWNSTAIRS CAFE INC		
15	28/10/2020 12:11	$41.00		NOOBER EATS		
16	28/10/2020 12:43		$15.00		ROBBIE Q.	Pay back for lunch

Jeremiah’s bank now offers a service that provides a tab separated output of each week’s bank statement automatically. Jeremiah would like to do some rudimentary investigations on his weekly spending and has asked you to do this in C# since he’s a real language purist and gets weird about these sorts of things.

He’s provided the output in the file: **qbank-export-1-11-2020.txt**

You can use any C# compiler and packages you wish to use for these task, but any nuget imports that Jeremiah has to install should be clearly indicated.

Jeremiah typically uses Microsoft Visual Studio 2016 and .NET Framework v4.7.x or above.

Console input and output is fine for all tasks, no front end is expected.

A sample structure and output for the program can be found on the next page.

TASK 1

Read the input data into your program from the output file and map the data into a class of your design. Reading ahead and seeing what other tasks you’ll need to perform may help influence your class design. Jeremiah will be running these operations once weekly, by loading the files in.

Identify the method that does this followed by any instructions to run this with a comment,
`//TASK 1- <<Instructions>>`

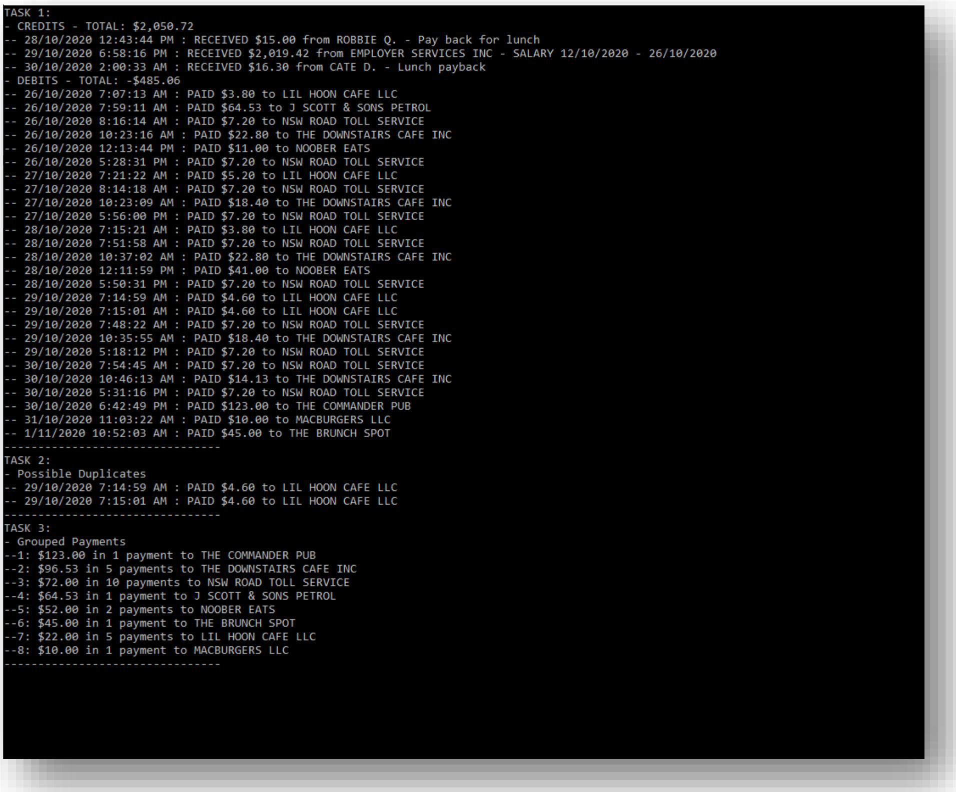
TASK 2

Jeremiah thinks that he may have been charged twice for coffee on the 29th of October. Write a method that can identify and print to the console any potential duplicates within the dataset.
Identify the method that does this followed by any instructions to run this with a comment,
`//TASK 2 - <<Instructions>>`

TASK 3

Jeremiah wants to know where he’s spending all his money. Output a ranked list of the combined total of every payee that Jeremiah has been spending money on from most expenditure to least.
Identify the method that does this followed by any instructions to run this with a comment,
`//TASK 3 - <<Instructions>>`

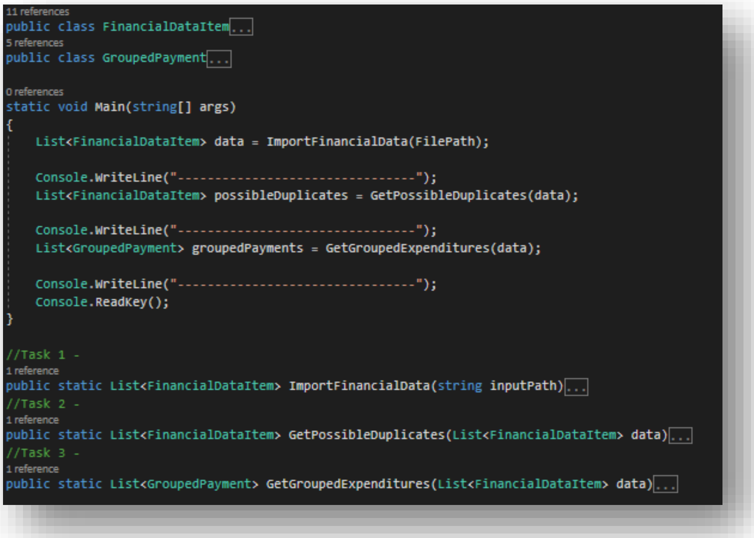
SAMPLE OUTPUT



```
TASK 1:
- CREDITS - TOTAL: $2,050.72
-- 28/10/2020 12:43:44 PM : RECEIVED $15.00 from ROBBIE Q. - Pay back for lunch
-- 29/10/2020 6:58:16 PM : RECEIVED $2,019.42 from EMPLOYER SERVICES INC - SALARY 12/10/2020 - 26/10/2020
-- 30/10/2020 2:00:33 AM : RECEIVED $16.30 from CATE D. - Lunch payback
- DEBITS - TOTAL: -$485.06
-- 26/10/2020 7:07:13 AM : PAID $3.80 to LIL HOON CAFE LLC
-- 26/10/2020 7:59:11 AM : PAID $64.53 to J SCOTT & SONS PETROL
-- 26/10/2020 8:16:14 AM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 26/10/2020 10:23:16 AM : PAID $22.80 to THE DOWNSTAIRS CAFE INC
-- 26/10/2020 12:13:44 PM : PAID $11.00 to NOOBER EATS
-- 26/10/2020 5:28:31 PM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 27/10/2020 7:21:22 AM : PAID $5.20 to LIL HOON CAFE LLC
-- 27/10/2020 8:14:18 AM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 27/10/2020 10:23:09 AM : PAID $18.40 to THE DOWNSTAIRS CAFE INC
-- 27/10/2020 5:56:00 PM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 28/10/2020 7:15:21 AM : PAID $3.80 to LIL HOON CAFE LLC
-- 28/10/2020 7:51:58 AM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 28/10/2020 10:37:02 AM : PAID $22.80 to THE DOWNSTAIRS CAFE INC
-- 28/10/2020 12:11:59 PM : PAID $41.00 to NOOBER EATS
-- 28/10/2020 5:50:31 PM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 29/10/2020 7:14:59 AM : PAID $4.60 to LIL HOON CAFE LLC
-- 29/10/2020 7:15:01 AM : PAID $4.60 to LIL HOON CAFE LLC
-- 29/10/2020 7:48:22 AM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 29/10/2020 10:35:55 AM : PAID $18.40 to THE DOWNSTAIRS CAFE INC
-- 29/10/2020 5:18:12 PM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 30/10/2020 7:54:45 AM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 30/10/2020 10:46:13 AM : PAID $14.13 to THE DOWNSTAIRS CAFE INC
-- 30/10/2020 5:31:16 PM : PAID $7.20 to NSW ROAD TOLL SERVICE
-- 30/10/2020 6:42:49 PM : PAID $123.00 to THE COMMANDER PUB
-- 31/10/2020 11:03:22 AM : PAID $10.00 to MACBURGERS LLC
-- 1/11/2020 10:52:03 AM : PAID $45.00 to THE BRUNCH SPOT
--------------------------------
TASK 2:
- Possible Duplicates
-- 29/10/2020 7:14:59 AM : PAID $4.60 to LIL HOON CAFE LLC
-- 29/10/2020 7:15:01 AM : PAID $4.60 to LIL HOON CAFE LLC
--------------------------------
TASK 3:
- Grouped Payments
--1: $123.00 in 1 payment to THE COMMANDER PUB
--2: $96.53 in 5 payments to THE DOWNSTAIRS CAFE INC
--3: $72.00 in 10 payments to NSW ROAD TOLL SERVICE
--4: $64.53 in 1 payment to J SCOTT & SONS PETROL
--5: $52.00 in 2 payments to NOOBER EATS
--6: $45.00 in 1 payment to THE BRUNCH SPOT
--7: $22.00 in 5 payments to LIL HOON CAFE LLC
--8: $10.00 in 1 payment to MACBURGERS LLC
--------------------------------
```

Note: You don't have to display your results the same way! Display them however you deem best.

SAMPLE STRUCTURE



```
11 references
public class FinancialDataItem...
5 references
public class GroupedPayment...

0 references
static void Main(string[] args)
{
    List<FinancialDataItem> data = ImportFinancialData(FilePath);

    Console.WriteLine("--------------------------------");
    List<FinancialDataItem> possibleDuplicates = GetPossibleDuplicates(data);

    Console.WriteLine("--------------------------------");
    List<GroupedPayment> groupedPayments = GetGroupedExpenditures(data);

    Console.WriteLine("--------------------------------");
    Console.ReadKey();
}

//Task 1 -
1 reference
public static List<FinancialDataItem> ImportFinancialData(string inputPath)...
//Task 2 -
1 reference
public static List<FinancialDataItem> GetPossibleDuplicates(List<FinancialDataItem> data)...
//Task 3 -
1 reference
public static List<GroupedPayment> GetGroupedExpenditures(List<FinancialDataItem> data)...
```

To get you started, you can start with this structure with the structure in the attached solution if you wish – or you can find a version of the program's main structure here: https://pastebin.com/K7LEDMch