```
class The_Bill():
    def __init__(self, n, m):
        self.n = n #Value or size of the group
        self.m = m #Value or size of the transaction

        self.balance = [0 for i in range(self.n)] #index depending on the inputted value of n
        self.transaction_balance = [] #blank list
        self.j = 0

    def Transaction(self):
        for y in range(self.m):
            self.transaction_id = input("Transaction ID: ") #A string to represent the Transaction ID
            self.paid_by = int(input("How many person paid: ")) #The size of how many person will pay
            self.split_as = int(input("How many person will split: ")) #The size of how many person will split

            for i in range(self.paid_by): #for loop for paid_by
                user_id,amount = map(int,input().split()) #Will ask for 2 inputs separated by a white space
                self.balance[user_id - 1] -= amount #Will determine the index in the list by subtracting it by 1

            for i in range(self.split_as): #for loop for split_as
                user_id,amount = map(int,input().split())
                self.balance[user_id - 1] += amount #Will determine the index in the list by subtracting it by 1

    def Bill_checker(self):
        print("\nOutput: \n")
        for i in range(self.n): #for loop based on the inputted value of n
            if(self.balance[i] > 0):
                self.current_balance = self.balance[i] #gets the value of balance same to the current balance
                while(self.current_balance > 0 and self.j < self.n): #checker if the current balance is higher than 0 and if the j is lower t
                    if(self.balance[self.j]>=0):  #if the index j of balance is higher equal than 0 it will increment the index j
                        self.j = self.j+1;
                        continue;
                    self.minimum_balance = min(self.current_balance, abs(self.balance[self.j])) #looks for the minimum balance and turns the
                    self.current_balance -= self.minimum_balance #subtracts the current balance with the min balance and assigns the result b
                    self.balance[self.j] += self.minimum_balance #Adds the balance with the min balance and assigns the result back to the ba
                    self.transaction_balance.append([[i+1, self.j+1], self.minimum_balance]) #Increment the list of transaction balance index

        for i in range(len(self.transaction_balance)): #for loop depending to the value of transaction balance list
            print("Person",self.transaction_balance[i][0][0],"owes","Person",
                  self.transaction_balance[i][0][1], self.transaction_balance[i][1]) #Outputs of persons that owes money

n = int(input("Enter a number for the size of the Group: ")) #Asks the user input for the size of the Group
m = int(input("Enter a number for the size of transaction: ")) #Asks the user input on how many transaction will there be
Jomey = The_Bill(n, m) #Assigning the class name to Jomey
Jomey.Transaction() #Call out of the transaction function
Jomey.Bill_checker() #Call out of the Bil checker function
```

```
Enter a number for the size of the Group: 6
Enter a number for the size of transaction: 5
Transaction ID: itsmylife
How many person paid: 2
How many person will split: 3
1 25
3 15
4 10
5 25
6 5
Transaction ID: itsnow
How many person paid: 1
How many person will split: 4
4 100
1 25
2 25
3 25
4 25
Transaction ID: ornever
How many person paid: 2
How many person will split: 2
5 30
3 10
1 25
4 15
Transaction ID: iaintgonna
How many person paid: 1
How many person will split: 3
2 150
1 50
```

```
2 50
3 50
Transaction ID: liveforever
How many person paid: 2
How many person will split: 2
5 13
6 25
4 25
1 13

Output:

Person 1 owes Person 2 75
Person 1 owes Person 4 13
Person 3 owes Person 4 12
Person 3 owes Person 5 18
Person 3 owes Person 6 20
```