You are given the following skeleton file and your job is to edit it according to the instructions below and in the .py file.

**File:**

[section2.py](https://psu.instructure.com/courses/2171603/files/134959110?wrap=1)[Download section2.py](https://psu.instructure.com/courses/2171603/files/134959110/download?download_frd=1)

**Task:**

You work at the grocery store as a data manager and you'd written a class GroceryStore that maintains everything. Your today's task is to do the following:

1. Complete **print\_vars().**This function doesn't take any external parameter, nor returns anything. In the body of the function, it simply prints the properties/variables of the object.
2. Complete **reassign\_Sales().**This function also doesn't take any external parameter, nor returns anything. Within the body, it simply reassigns the value of sales to 0 if its actual value is greater than or equal to 80.
3. Complete **switch\_branch().**This function doesn't take any external parameter, nor returns anything. Within the body of the function, do this:
   * If the first element of branches is 0, assign the second element to the variable named name
   * If the first element of branches is 1, assign the third element to the variable named name
4. Complete **increase\_price().**This function takes an external parameter named amount, but doesn't return anything. Within the body, it simply goes over the list of prices and add amount to each price.
5. Complete **check\_daily\_profit().**This function takes an external parameter named marketProfit. Within the body of the function, it does the following:
   * If your profit is negative or equal to 0, return: Company is in loss (Don't print it)
   * If your profit is greater than or equal to marketProfit, return: Profit is good for today
   * If your profit is greater than 0 and less than marketProfit, then return: Profit is low for today
6. Invoke all functions according to the comments in the attached file.

**Code:**

class GroceryStore:

  def \_\_init\_\_(self, prices, name, sales, profit, branches):

    self.prices=prices #has a list of prices of various items in the store

    self.name=name #name of the grocery store

    self.sales=sales #Highest sale this month

    self.profit=profit #Everyday profit

    self.branches=branches #Tuple that has three elements

  def print\_vars(self):

    print(self.prices, self.name, self.sales, self.profit, self.branches)

  def reassign\_Sales(self):

    if self.sales>=80:

      self.sales=0

  def switch\_branch(self):

    if self.branches[0]==0:

      self.name=self.branches[1]

    elif self.branches[0]==1:

      self.name=self.branches[2]

  def increase\_price(self, increase):

    for i in range(len(self.prices)):

      self.prices[i]+=increase

  def check\_daily\_profit(self, priceVal):

    if self.profit<priceVal and self.profit>0:

      return "Profit for today was too low"

    elif self.profit>=priceVal:

      return "We made a very good profit today"

    elif self.profit<=0:

      return "Company is in loss"

#Do not change initial parameters

prices=[100, 23, 45, 66, 77, 89, 00]

name="Walmart"

sales=95

profit=45.44

branch=(0, "Walmart 1441", "Walmart 3477")

gross=GroceryStore(prices, name,sales, profit, branch)

gross.print\_vars()

gross.reassign\_Sales()

gross.print\_vars()

gross.switch\_branch()

gross.print\_vars()

gross.increase\_price(30)

gross.print\_vars()

print(gross.check\_daily\_profit(100))