Explanation of the Steps taken in the assignment

Ahmed Ashraf Mohamed

1 Loading The data

Import the numpy library and load the sales data into a variable `data` using function `numpy.load()`

```
import numpy as np
data = np.load("sales_data.npy")
```

Check the sales data, its dimensions and shape of the numpy array. assign the number of rows of the data into a variable `rows` and the number of columns into a variable `cols`

```
print(data,data.ndim,data.shape,sep="\n")
rows = data.shape[0]
cols = data.shape[1]
```

2 Slice the array to extract the sales data for the year 2022.

Since the data is arranged, we can slice the numpy array easily. Knowing that the data contains five months of each year.

```
data2022 = data[10:15,3]
print(data2022)
```

3 Calculate the total sales amount for each country for the year 2022.

Since the the total sales amount is the fourth column, we can loop through the 2022 slice or through the whole data, and extract the total sales amount for the year 2022 of each country, and calculate the total sales amount for all the countries in 2022

```
totalCountries = 0 # used to hold the total sales for all countries
# looping through the whole data using the rows variable
```

```
for i in range(rows):
    # checking the for year 2022
    if "2022" in data[i,0]:
        print("Country:", data[i,1], "Total Sales:", data[i,4],sep = "\t", end="\n")
        totalCountries += int(data[i,4])
print("Total Sales for all Countries in the year 2022:",totalCountries)
```

4 Calculate the mean sales amount for each product over the entire period.

We can create a list for each product, knowing that the data contains only three products, and looping through the whole data and checking for the product to append the sales in the products list. Then convert the lists into numpy arrays to take advantage of the mean method in the numpy arrays.

```
productASales = []
productBSales = []
productCSales = []
for i in range(rows):
    if data[i,2] == "Product A":
        productASales.append(int(data[i,4]))
    elif data[i,2] == "Product B":
        productBSales.append(int(data[i,4]))
    else:
        productCSales.append(int(data[i,4]))
productASales = np.array(productASales)
productBSales = np.array(productBSales)
productCSales = np.array(productCSales)
print(productASales,productBSales,productCSales,sep="\n")
print("Product A mean Sales:", productASales.mean())
print("Product B mean Sales:", productBSales.mean())
print("Product C mean Sales:", productCSales.mean())
```

5 Extract the total sales amount for each country for the year 2023 and Append it to a separate array.

There are two different solutions for this problem

5.1 First solution: slicing

Slicing will help us avoid looping through the data and being faster, since we know that the data is arranged.

```
data2023 = data[15:,(1,4)]
print(data2023)
```

5.2 Second Solution: Looping

creating an empty list, and looping through the whole data to identify the rows of the year 2023 and append the data into a list. then convert the list into a numpy array

```
data2023 = []
for i in range(rows):
    if '2023' in data[i,0]:
        data2023.append(data[i])
data2023 = np.asarray(data2023)
print(data2023[:,(1,4)])
```

6 Delete the sales data for Ireland for the year 2021

To not change the original data, we can make a copy using `ndarray.copy()` function and then loop through the new data to identify the specified row to delete

```
newData = data.copy()
for i in range(rows):
    if "2021" in data[i,0] and "Ireland" in data[i,1]:
        newData = np.delete(newData,i,axis=0)
print(newData)
```

7 Find the unique product names and print them sorted alphabetically.

using the `np.unique()` function to identify unique elements in the array. and sorted function to sort them alphabetically

```
print(sorted(np.unique(data[:,2])))
```

8 Create a 2D array with random values of shape (20, 3) and concatenate it with the sales data array along the second axis.

To create random array with the shape `(20,3)`, we are going to use `np.random.randint()` and specify the size, or rather the shape, of the array as one the arguments of the function. And then using the `np.concatenate()` function we are going to concatenate it with the original data along the second axis `axis = 1`

```
randArr = np.random.randint(0,2,size=(20,3))
concatData = np.concatenate((randArr,data),axis = 1)
print(concatData)
```

9 Save the output in modified sales data.npy and Explain what happened to the matrix by doing the concatenation above.

Using the `numpy.save()` function to save the numpy array as an external file with the extension `.npy`

```
np.save('modified_sales_data.npy',concatData)
```