CSV Exercise

1.

```
import matplotlib.pyplot as plt
import numpy as np
from math import ceil
import csv

daysSteps : dict = {}
daysDivider : dict ={}
dayArray = []
barNames = []
#togetMedian
daysStepsArray : dict ={}

f = open('activity.csv','r')
lines = f.readlines()
lines.pop(0)
def initializeArrays():
    for i in lines:
        i = i.replace('"',"")
        i = i.replace("\n","")
        i = i.split(',')

        if(i[1] not in daysSteps):
            if(i[0] != 'NA'):
                daysSteps[i[1]] = int(i[0])
                daysDivider[i[1]] = 1
                daysStepsArray[i[1]] = [i[0]]
        else:
            if(i[0] != 'NA'):
                daysSteps[i[1]] += int(i[0])
                daysDivider[i[1]] += 1
                daysStepsArray[i[1]].append(i[0])

initializeArrays()
keys = list(daysSteps.keys())
bars = int(input("Enter number of bars :"))
#change the magic number to display more or less bars
for i in range(bars):
    dayArray.append(daysSteps[keys[i]])
    barNames.append(keys[i])

for i in range(bars):
    print("Median of day " + str(keys[i]) + " : ", end ="")
    if(len(daysStepsArray[keys[i]])%2 == 1):
        daysStepsArray[keys[i]].sort()
        medianLocation = ceil(len(daysStepsArray[keys[i]]))/2
        print(daysStepsArray[keys[medianLocation]])

    else:
        daysStepsArray[keys[i]].sort()
        medianLocation =  ceil(len(daysStepsArray[keys[i]]) / 2)
        print(float(int(daysStepsArray[keys[i]][medianLocation]) + int(daysStepsArray[keys[i]][medianLocation+1]) / 2))

    print("Means of day " + str(keys[i]) + " : ", end = "")
    print(daysSteps[keys[i]] / daysDivider[keys[i]] , "\n")

plt.bar(barNames,dayArray)
plt.show()
```

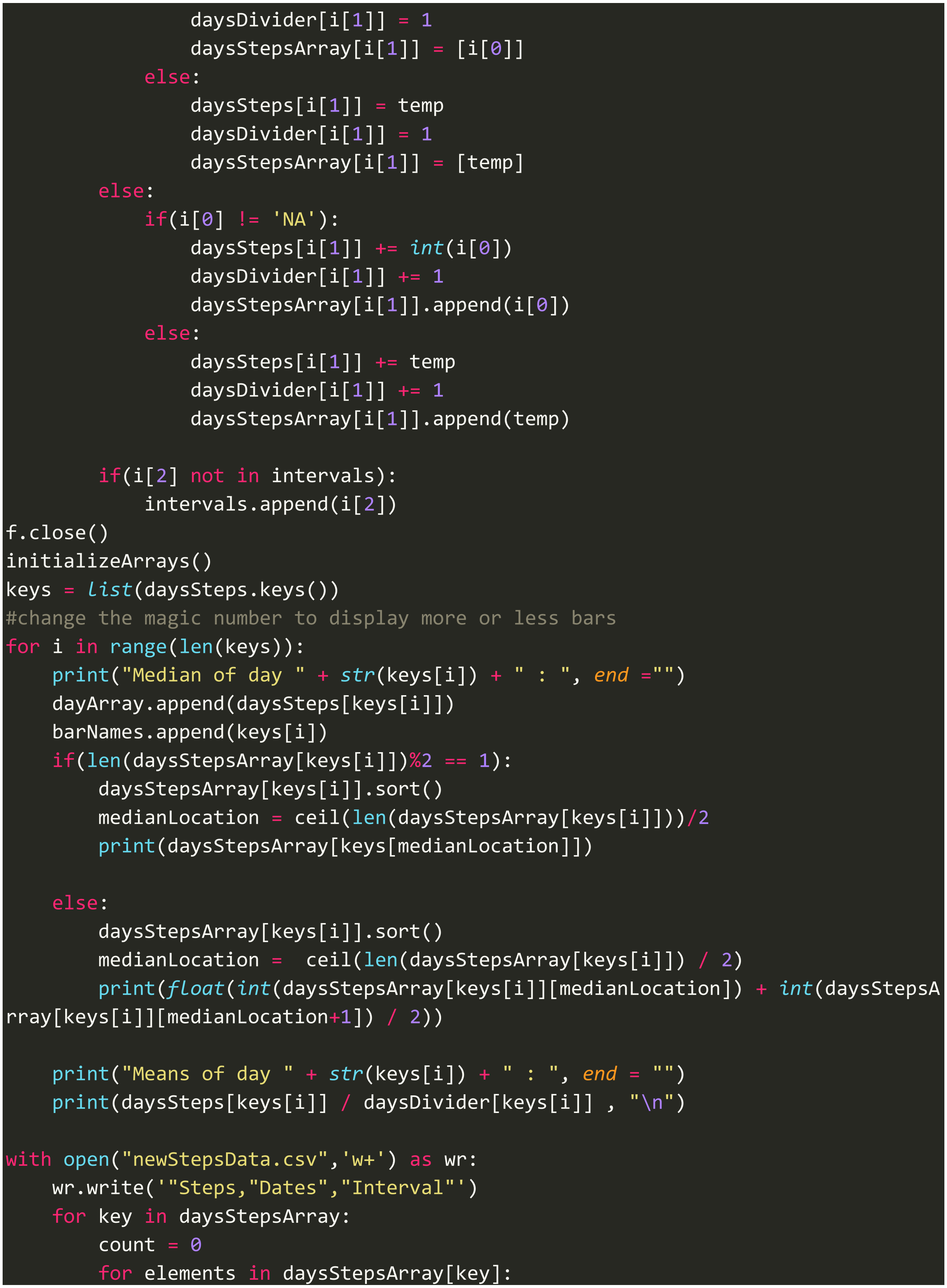
2.

```
import matplotlib.pyplot as plt
import numpy as np
from math import ceil
import csv

intervalsAndSums : dict = {}
totalDays = 61
f = open('activity.csv','r')
lines = f.readlines()
lines.pop(0)
def initializeArrays():
    for i in lines:
        i = i.replace('"',"")
        i = i.replace("\n","")
        i = i.split(',')

        if(i[0] != "NA"):
            if(str(i[2]) not in intervalsAndSums):
                intervalsAndSums[str(i[2])] = int(i[0])
            else:
                intervalsAndSums[str(i[2])] += int(i[0])
f.close()
```

```
initializeArrays()

intervalKeys = list(intervalsAndSums.keys())
intervalValues = []
for a in intervalsAndSums:
    intervalValues.append(intervalsAndSums[a]/61)
print("The interval with maximum steps is : ")
maxValue = max(intervalValues)
print(intervalsAndSums)
print("Interval : " + intervalKeys[intervalValues.index(maxValue)])
print("Averaged Step : " + str(maxValue))
plt.bar(intervalKeys,intervalValues)
plt.xlabel("Intervals")
plt.ylabel("Averaged Steps")
plt.show()
```

3.

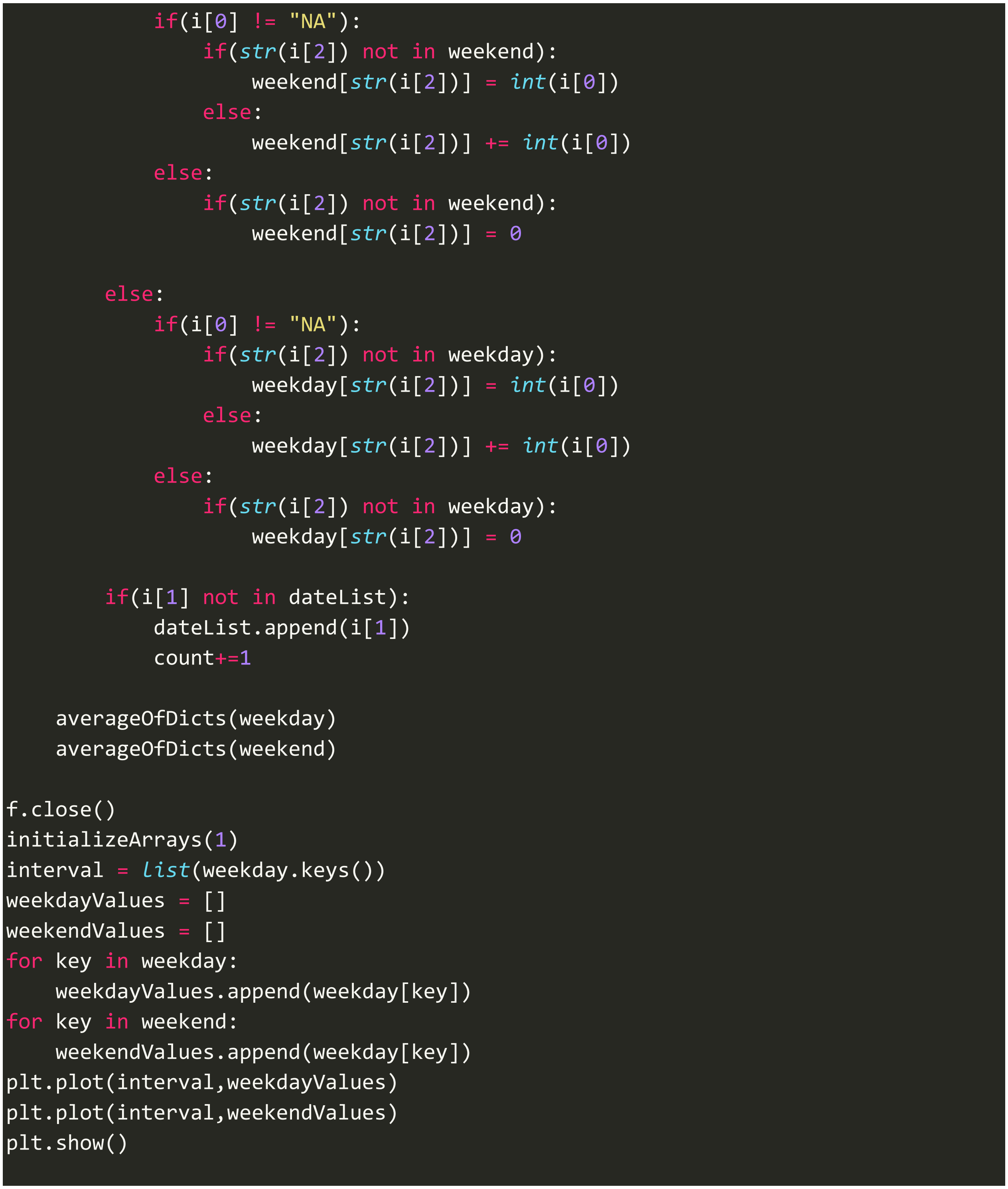
```
import matplotlib.pyplot as plt
import numpy as np
from random import randint
from math import ceil
import csv

daysSteps : dict = {}
daysDivider : dict ={}
dayArray = []
barNames = []
intervals = []

#togetMedian
daysStepsArray : dict ={}

f = open('activity.csv','r')
lines = f.readlines()
lines.pop(0)
def initializeArrays():
    for i in lines:
        i = i.replace('"',"")
        i = i.replace("\n","")
        i = i.split(',')
        temp = randint(0,300)
        if(i[1] not in daysSteps):
            if(i[0] != 'NA'):
                daysSteps[i[1]] = int(i[0])
```



```
                daysDivider[i[1]] = 1
                daysStepsArray[i[1]] = [i[0]]
            else:
                daysSteps[i[1]] = temp
                daysDivider[i[1]] = 1
                daysStepsArray[i[1]] = [temp]
        else:
            if(i[0] != 'NA'):
                daysSteps[i[1]] += int(i[0])
                daysDivider[i[1]] += 1
                daysStepsArray[i[1]].append(i[0])
            else:
                daysSteps[i[1]] += temp
                daysDivider[i[1]] += 1
                daysStepsArray[i[1]].append(temp)

        if(i[2] not in intervals):
            intervals.append(i[2])
f.close()
initializeArrays()
keys = list(daysSteps.keys())
#change the magic number to display more or less bars
for i in range(len(keys)):
    print("Median of day " + str(keys[i]) + " : ", end ="")
    dayArray.append(daysSteps[keys[i]])
    barNames.append(keys[i])
    if(len(daysStepsArray[keys[i]])%2 == 1):
        daysStepsArray[keys[i]].sort()
        medianLocation = ceil(len(daysStepsArray[keys[i]]))/2
        print(daysStepsArray[keys[medianLocation]])

    else:
        daysStepsArray[keys[i]].sort()
        medianLocation =  ceil(len(daysStepsArray[keys[i]]) / 2)
        print(float(int(daysStepsArray[keys[i]][medianLocation]) + int(daysStepsArray[keys[i]][medianLocation+1]) / 2))

    print("Means of day " + str(keys[i]) + " : ", end = "")
    print(daysSteps[keys[i]] / daysDivider[keys[i]] , "\n")

with open("newStepsData.csv",'w+') as wr:
    wr.write('"Steps,"Dates","Interval"')
    for key in daysStepsArray:
        count = 0
        for elements in daysStepsArray[key]:
            wr.write(str(elements)+","+key+","+intervals[count]+"\n")
            count+=1
            if(count == len(intervals)):
                count=0


plt.bar(barNames,dayArray)
plt.show()
```

4.

```
import matplotlib.pyplot as plt
import numpy as np
from random import randint
from math import ceil
import csv

weekday : dict = {}
weekend : dict = {}
averagedWeekday : dict
averagedWeekend : dict

# used to determine whether a day is a weekend or weekday
# weekend is determined by modulo 7 or 8 (saturday and sunday)
count = 1

f = open('activity.csv','r')
dateList =[]
lines = f.readlines()
lines.pop(0)
def averageOfDicts(targetDict):
    totalDay = count
    for i in targetDict:
        targetDict[i] = targetDict[i] / totalDay

def initializeArrays(count):
    for i in lines:
        i = i.replace('"',"")
        i = i.replace("\n","")
        i = i.split(',')

        if((count % 6) == 0 or (count % 7) == 0):
```



```
            if(i[0] != "NA"):
                if(str(i[2]) not in weekend):
                    weekend[str(i[2])] = int(i[0])
                else:
                    weekend[str(i[2])] += int(i[0])
            else:
                if(str(i[2]) not in weekend):
                    weekend[str(i[2])] = 0

        else:
            if(i[0] != "NA"):
                if(str(i[2]) not in weekday):
                    weekday[str(i[2])] = int(i[0])
                else:
                    weekday[str(i[2])] += int(i[0])
            else:
                if(str(i[2]) not in weekday):
                    weekday[str(i[2])] = 0

        if(i[1] not in dateList):
            dateList.append(i[1])
            count+=1

    averageOfDicts(weekday)
    averageOfDicts(weekend)

f.close()
initializeArrays(1)
interval = list(weekday.keys())
weekdayValues = []
weekendValues = []
for key in weekday:
    weekdayValues.append(weekday[key])
for key in weekend:
    weekendValues.append(weekday[key])
plt.plot(interval,weekdayValues)
plt.plot(interval,weekendValues)
plt.show()
```