Introduction to Data Science – Session 2 – Solutions

Exercise-1: S2-Ex1 ("if" and "for" statements)

```
ageList = [5, 11, 17, 18, 27, 43, 55, 65, 67]
for i in ageList:
     if i < 18:
          canVote = 'No'
     else:
          canVote = 'Yes'
     print (i, canVote)
```

Exercise-1: S2-Ex1-2 ("if" and "for" statements)

```
ageList = [5, 11, 17, 18, 27, 43, 55, 65, 67]
noOfChildren = 0
noOfTeenagers = 0
noOfAdults = 0
noOfSrAdults = 0
for i in ageList:
    if (i <= 12):
        noOfChildren += 1
    elif ((i >= 13) and (i <= 17)):
        noOfTeenagers += 1
    elif ((i >= 18) and (i <= 59)):
        noOfAdults += 1
    elif (i >= 60):
        noOfSrAdults += 1
    else:
        print ("Invalid data")
print ("No. of Children", noOfChildren)
print ("No. of Teenagers", noOfTeenagers)
print ("No. of Adults", noOfAdults)
print ("No. of Sr. Adults", noOfSrAdults)
```

Exercise-2: S2-Ex2 ("if" and "for" statements)

```
marksList = [7, 11, 29, 30, 50, 57, 69, 75, 88, 89, 90, 92, 97]
noOfAs = 0
noOfBs = 0
noOfCs = 0
noOfDs = 0
```

```
noOfEs = 0
noOfFs = 0
gradeList = []
for marks in marksList:
    if (marks < 10):
        noOfFs += 1
        gradeList.append ('F')
    elif ((marks >= 10) and (marks <= 29)):
        noOfEs += 1
        gradeList.append ('E')
    elif ((marks >= 30) and (marks <= 49)):
        noOfDs += 1
        gradeList.append ('D')
    elif ((marks >= 50) and (marks <= 69)):
        noOfCs += 1
        gradeList.append ('C')
    elif ((marks >= 70) and (marks <= 89)):
        noOfBs += 1
        gradeList.append ('B')
    elif ((marks >= 90) and (marks <= 100)):
        noOfAs += 1
        gradeList.append ('A')
    else:
        print ("Invalid data")
print ("No. of As", noOfAs)
print ("No. of Bs", noOfBs)
print ("No. of Cs", noOfCs)
print ("No. of Ds", noOfDs)
print ("No. of Es", noOfEs)
print ("No. of Fs", noOfFs)
print (gradeList)
```

Exercise-3: S2-Ex3 ("while" statement)

```
#Sums expenses for Jan-Mar
expenseList = [509.50, 1019.43, 1527.22]
i = 0
totalExpense = 0
while i < 3:
    totalExpense = totalExpense + expenseList [i]
    i = i + 1
print (totalExpense)
```

Exercise-3: S2-Ex3-2 ("while" statement)

```
#Sums goals
goalsList = [0, 2, 1, 0, 3, 2, 3]
i = 0
totalGoals = 0
while i in range (len (goalsList)):
    totalGoals += goalsList [i]
    i = i + 1
print (totalGoals)
```

Exercise-4: S2-Ex4 (Binary Search)

```
idList = (2, 5, 10, 17, 20, 29, 33, 49, 51)
searchForItem = 51 # Try different items
i = 0
j = len (idList) - 1
found = False
# print (len (idList), i, j)
while i <= j:
    k = (i + j)//2
#   print (i, j, k)
    if idList [k] < searchForItem:
        i = k + 1
#       print (i, j, k)
    elif idList [k] > searchForItem:
        j = k - 1
#       print (i, j, k)
    else:
#       print (i, j, k)
        found = True
        break
if (found == True):
    print ('Item found at position: ', k)
#   print (i, j, k)
else:
    print ('Item not found')
```

Exercise-5: S2-Ex5 (Simple Bar Graph)

```
import matplotlib
import matplotlib.pyplot as plt
```

```
y = (17, 20, 15, 17, 15)
x = ('Class 1', 'Class 2', 'Class 3', 'Class 4', 'Class 5')
plt.xlabel ('Class')
plt.ylabel ('Attendance')
plt.title ('Class Attendance')
plt.bar (x, y)
```

Exercise-6: S2-Ex6 (Bubble Sort)

```
#Sorted marksList = [7, 11, 29, 30, 50, 57, 69, 75, 88, 89]
marksList = [30, 50, 11, 7, 57, 88, 75, 89, 69, 29]

lenML = len (marksList)
lenSort = lenML - 1
sortedFlag = False
while (not sortedFlag):
    lenSort = lenML - 1
    sortedFlag = True
    for i in range (lenSort):
        if (marksList [i] > marksList [i + 1]):
            temp = marksList [i]
            marksList [i] = marksList [i + 1]
            marksList [i + 1] = temp
            sortedFlag = False
    print (marksList)
```