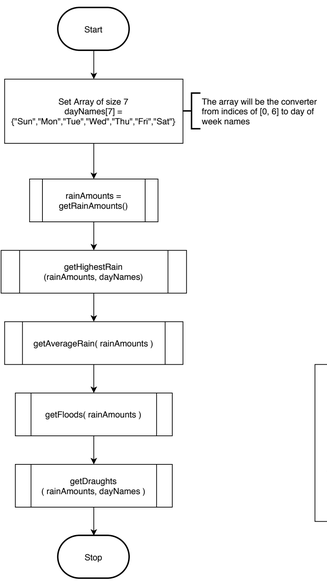
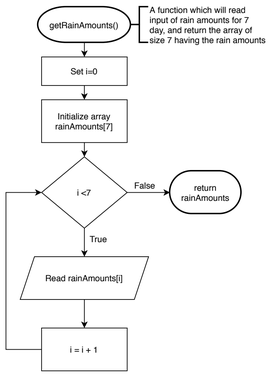
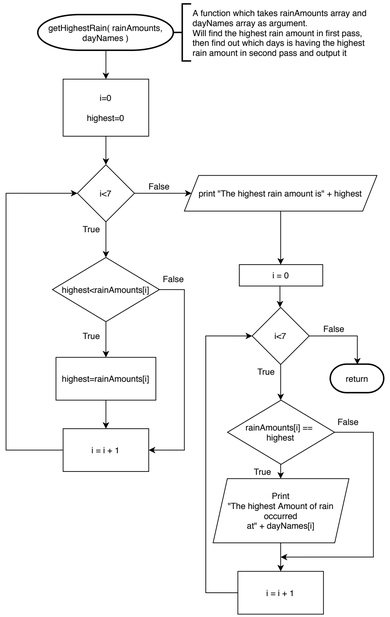
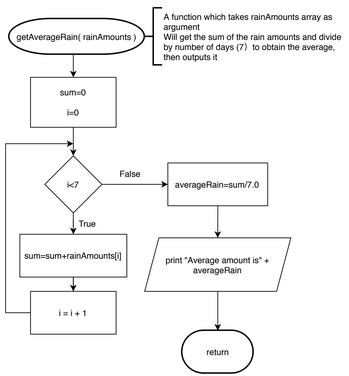
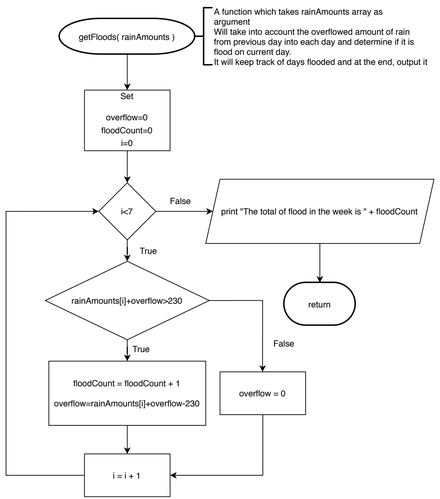
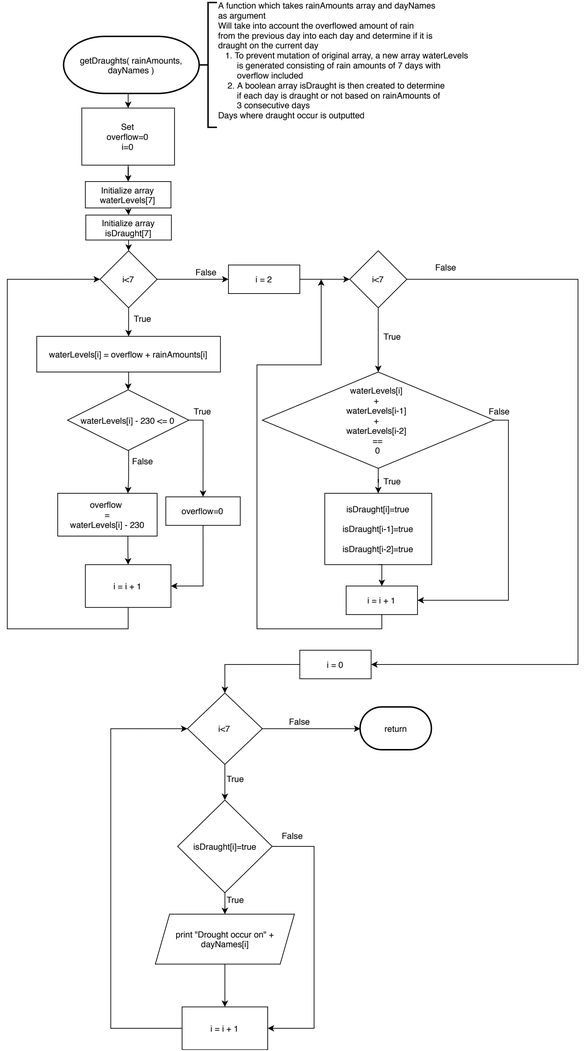

Start
Set Array of size 7
dayNames[7] =
{"Sun","Mon","Tue","Wed","Thu","Fri","Sat"}
The array will be the converter from indices of [0, 6] to day of week names
rainAmounts = getRainAmounts()
getHighestRain (rainAmounts, dayNames)
getAverageRain(rainAmounts)
getFloods(rainAmounts)
getDraughts (rainAmounts, dayNames)
Stop


getRainAmounts()
A function which will read input of rain amounts for 7 day, and return the array of size 7 having the rain amounts
Set i=0
Initialize array rainAmounts[7]
i <7
False
return rainAmounts
True
Read rainAmounts[i]
i = i + 1


getHighestRain(rainAmounts, dayNames)
A function which takes rainAmounts array and dayNames array as argument.
Will find the highest rain amount in first pass, then find out which days is having the highest rain amount in second pass and output it
i=0
highest=0
i<7
False
print "The highest rain amount is" + highest
True
highest<rainAmounts[i]
False
True
highest=rainAmounts[i]
i = i + 1
i = 0
i<7
False
return
True
rainAmounts[i] == highest
False
True
Print "The highest Amount of rain occurred at" + dayNames[i]
i = i + 1


getAverageRain(rainAmounts)
A function which takes rainAmounts array as argument
Will get the sum of the rain amounts and divide by number of days (7) to obtain the average, then outputs it
sum=0
i=0
i<7
False
averageRain=sum/7.0
True
sum=sum+rainAmounts[i]
print "Average amount is" + averageRain
i = i + 1
return


getFloods(rainAmounts)
A function which takes rainAmounts array as argument
Will take into account the overflowed amount of rain from previous day into each day and determine if it is flood on current day.
It will keep track of days flooded and at the end, output it
Set
overflow=0
floodCount=0
i=0
i<7
False
print "The total of flood in the week is " + floodCount
True
return
rainAmounts[i]+overflow>230
False
True
floodCount = floodCount + 1
overflow=rainAmounts[i]+overflow-230
overflow = 0
i = i + 1


getDraughts(rainAmounts, dayNames)
A function which takes rainAmounts array and dayNames as argument
Will take into account the overflowed amount of rain from the previous day into each day and determine if it is draught on the current day
1. To prevent mutation of original array, a new array waterLevels is generated consisting of rain amounts of 7 days with overflow included
2. A boolean array isDraught is then created to determine if each day is draught or not based on rainAmounts of 3 consecutive days
Days where draught occur is outputted
Set
overflow=0
i=0
Initialize array waterLevels[7]
Initialize array isDraught[7]
i<7
False
i = 2
i<7
False
True
waterLevels[i] = overflow + rainAmounts[i]
True
waterLevels[i] + waterLevels[i-1] + waterLevels[i-2] == 0
False
waterLevels[i] - 230 <= 0
True
False
overflow = waterLevels[i] - 230
overflow=0
isDraught[i]=true
isDraught[i-1]=true
isDraught[i-2]=true
i = i + 1
i = i + 1
i = 0
i<7
False
return
True
isDraught[i]=true
False
True
print "Drought occur on" + dayNames[i]
i = i + 1