

Handout 4

Reminder on the record type that's holding the data we need

```
RECORD OutletData IS {  
    STRING businessName,    # pos 5  
    STRING businessType,    # pos 6  
    STRING postcode,        # pos 10  
    STRING ratingDate,      # pos 13  
    STRING ratingValue,     # pos 11  
    REAL longitude,         # pos 21  
    REAL latitude           # pos 22  
}
```

Reminder on this data structure

DECLARE outlets INITIALLY <the data that's already been read in>

Program 1: Count up how many outlets passed in the G12 area

Function to check whether the postcode areas of two

postcodes match

Assumes all postcode areas fit into three characters -

e.g. G4 will be passed as "G4 "

FUNCTION postCodeAreaMatch(STRING pc1, STRING pc2) RETURNS BOOLEAN

RETURN pc1[0] = pc2[0] AND

pc1[1] = pc2[1] AND

pc1[2] = pc2[2]

END FUNCTION

The count

DECLARE passedInG12 INITIALLY 0

Now scan through all outlets, checking on rating status

FOR EACH outlet FROM outlets DO

IF outlet.ratingValue = "Pass" AND postCodeAreaMatch("G12", outlet.postcode) THEN

SET passedInG12 TO passedInG12 + 1

END IF

END FOR EACH

Now print out the result

SEND "Number of passing outlets in G12 is " TO DISPLAY

SEND passedInG12 TO DISPLAY

```

# -----
# Program 2: Which post-code area (e.g. G12, G4, etc) has the highest percentage of failed outlets at this
time?

# Strips off the first three characters, the postcode area, of the supplied postcode
FUNCTION getPCAreaText( STRING postcode ) RETURNS STRING
    # Concatenates the first three characters of the postcode passed in
    RETURN [ postcode[ 0 ] ] & [ postcode[ 1 ] ] & [ postcode[ 2 ] ]
END FUNCTION

RECORD PostcodeAreaData IS { STRING postcodeArea, INTEGER failedOutlets, INTEGER totalOutlets }

# This is the summarised data on failed outlets in each postcode area
DECLARE pcAreas AS ARRAY OF PostcodeAreaData INITIALLY []

# Given a postcode, returns the entry in pcAreas, if it exists, or makes a new one
# for this postcode
FUNCTION getPCArea( STRING postcode ) RETURNS PostcodeAreaData
    DECLARE pos INITIALLY 0
    DECLARE found INITIALLY false
    DECLARE thisPCAText INITIALLY getPCAreaText( postcode )

    WHILE pos < length( pcAreas ) DO
        IF pcAreas[ pos ].postcodeArea = thisPCAText THEN
            SET found TO true
        ELSE
            SET pos TO pos + 1
        END IF
    END WHILE

    IF found THEN
        RETURN pcAreas[ pos ]
    ELSE
        DECLARE newPCArea INITIALLY PostcodeAreaData( thisPCAText, 0, 0 )
        SET pcAreas TO pcAreas & [ newPCArea ]
        RETURN newPCArea
    END IF
END FUNCTION

# Traverse over every outlet
FOR EACH outlet FROM outlets DO
    DECLARE pcAreaData INITIALLY getPCArea( outlet.postcode )

    SET pcAreaData.totalOutlets TO pcAreaData.totalOutlets + 1

    IF outlet.ratingValue = "Improvement Required" THEN
        SET pcAreaData.failedOutlets TO pcAreaData.failedOutlets + 1
    END IF
END FOR EACH

# Now traverse over the summarised data to find the 'worst' postcode area
DECLARE worstPCA INITIALLY pcAreas[ 0 ]
DECLARE worstPercentage INITIALLY 100 * worstPCA.failedOutlets / worstPCA.totalOutlets

FOR EACH thisPCA FROM pcAreas DO
    DECLARE thisPercentage INITIALLY 100 * thisPCA.failedOutlets / thisPCA.totalOutlets
    IF thisPercentage > worstPercentage THEN
        SET worstPercentage TO thisPercentage
        SET worstPCA TO thisPCA
    END IF
END FOR EACH

SEND "The worst postcode area is " & worstPCA.postcodeArea TO DISPLAY

```

```
# -----  
# Program 3 – Get a count of all the failed outlets  
  
# Get file name and open the file  
SEND "Please type in the file name: " TO DISPLAY  
RECEIVE filename FROM (STRING) KEYBOARD  
OPEN filename  
  
DECLARE failedOutletCount INITIALLY 0  
  
RECEIVE nextLine FROM (STRING) filename  
WHILE nextLine != "." DO  
    DECLARE splitL INITIALLY split( nextLine, "," )  
  
    IF splitL[ 11 ] = "Improvement Required" THEN  
        SET failedOutletCount TO failedOutletCount + 1  
    END IF  
  
    RECEIVE nextLine FROM (STRING) filename  
END WHILE  
  
SEND "The total number of failed outlets is " TO DISPLAY  
SEND failedOutletCount TO DISPLAY
```