```
# Handout 2
#Haggis program to read specific elements of an FSIH scheme CSV file into records

# Dummy type conversion function from STRING to REAL
FUNCTION real( STRING stringToConvert ) RETURNS REAL
    <Code to do conversion>
END FUNCTION

# Dummy split function
FUNCTION split( STRING source, STRING delimiter ) RETURNS ARRAY OF STRING
    <Splits up 'source' into an array of substrings, using 'delimiter' to determine where to split>
END FUNCTION

# Record type to hold the data we need
# The position in a line in the file is given for each field
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
}

# Function to create a new outlet record from a line of data
FUNCTION makeOutlet( ARRAY OF STRING line ) RETURNS OutletData
    RETURN OutletData( line[ 5 ], line[ 6 ],
                       line[ 10 ], line[ 13 ],
                       line[ 11 ],
                       real( line[ 21 ] ), real( line[ 22 ] ) )
END FUNCTION

# Create the data structure to hold the entire dataset
DECLARE outlets AS ARRAY OF OutletData INITIALLY []

# Get file name and open the file
SEND "Please type in the file name: " TO DISPLAY
RECEIVE filename FROM (STRING) KEYBOARD
OPEN filename

# Read in the lines from the file, one at a time
# Assume there is a sentinel line with only a full stop on it
RECEIVE nextLine FROM (STRING) filename  # may not check correctly with parser yet
WHILE nextLine != "." DO
    DECLARE splitL INITIALLY split( nextLine, "," )

    # Create a new outlet and add to the collection
    DECLARE thisOutlet INITIALLY makeOutlet( splitL )
    SET outlets TO outlets & [ thisOutlet ]

    # Get the next line from the file
    RECEIVE nextLine FROM (STRING) filename
END WHILE
```