|  |  |  |
| --- | --- | --- |
| **Function signature** | | **check\_allowed\_char**(*input, type, config*) |
| This function is used to validate the allowed characters in the input. It returns true if the input has the valid format type (alphabetic, alphanumeric, numeric), otherwise it returns false. | | |
| Arguments: | * *Input:* input attribute (string) * *Type:* determines the type of input e.g., company name, email address, etc. * *Config*– reference to a JSON Object storing the types and patterns. | |
| Returns: | Boolean (True/False) | |
| Example: | Input type: Email address  check\_allowed\_char("sara1295@ohio.edu","email","conf1")  Returns: True | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **check\_ invalid\_char**(*input, type, config*) |
| This function is used to check the existence of the invalid characters in the input. It returns true if the input contains invalid characters or patterns, otherwise it returns false. | | |
| Arguments: | * *Input:* input attribute (string) * *Type:* determines the type of input e.g., company name, email address, etc. * *Config*– reference to a JSON Object storing the types and patterns. | |
| Returns: | Boolean (True/False) | |
| Example: | Input type: Shipping address  check\_invalid\_char( "2 Andover Rd. P.O. 25678", " shippingAddress", "conf1")  Returns: True | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **check\_req\_char** (*input, type, config*) |
| This function is used to check the existence of the required characters in the input. It returns true if the input contains the required characters, otherwise it returns false. It used in validating email address. | | |
| Arguments: | * *Input:* input attribute (string) * *Type:* determines the type of input e.g., company name, email address, etc. * *Config*– reference to a JSON Object storing the types and patterns. | |
| Returns: | Boolean (True/False) | |
| Example: | Input type: Email address  check\_req\_char("sa.com@ohio", "email", "conf1")  Returns: True | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **field\_length\_check** (*input, type, config*) |
| This function is used to check the length of the input. It returns true if the input has the specified length, otherwise it returns false. | | |
| Arguments: | * *Input:* input attribute (string) * *Type:* determines the type of input e.g., company name, email address, etc. * *Config*– reference to a JSON Object storing the types and patterns. | |
| Returns: | Boolean (True/False) | |
| Example: | Input type: Shipment mode  field\_length\_check( "2525", "mode", "conf1")  Returns: False | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **range\_val\_check (***input, type, config, lbeq=false, ubeq=false***)** |
| This function is used to validate the range of the input. It returns true if the value of input is in the range, otherwise it returns false. | | |
| Arguments: | * *Input:* input attribute (string) * *Type:* determines the type of input e.g., company name, email address, etc. * *Config*– reference to a JSON Object storing the types and patterns. * *Lbeq:* set to TRUE if the lower bound check is desired. * *ubeq:* set to TRUE if the upper bound check is desired. | |
| Returns: | Boolean (True/False) | |
| Example: | Input type: Shipment mode  range\_val\_check(“15”,"ship\_date\_month", "conf1", *lbeq=false, ubeq=false* )  Returns: False | |
| **Function signature** | | **field\_length\_check** (*input, type, config*) |
| This function is used to validate the length of the input. It returns true if the input length is correct, otherwise it returns false. | | |
| Arguments: | * *Input:* input attribute (string) * *Type:* determines the type of input e.g., company name, email address, etc. * *Config*– reference to a JSON Object storing the types and patterns. | |
| Returns: | Boolean (True/False) | |
| Example: | Input type: Ceased operation month  range\_val\_check( “15”,"ship\_date\_month", "conf1")  Returns: False | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **presence\_check(*input*)** |
| This function is used to determine the presence of the input. It returns true if input is available, otherwise it returns false. | | |
| Arguments: | * *Input:* input attribute (string) * *Type:* determines the type of input e.g., company name, email address, etc. * *Config*– reference to a JSON Object storing the types and patterns. | |
| Returns: | Boolean (True/False) | |
| Example: | Input type: Mailing city  Presence\_check("Boston")  Returns: True | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function signature** | **lkup\_exhaustive\_m**(*table, column, index*) | | | |
| This function is used to perform exhaustive search and find all matching values from a lookup table. The function returns an object with two attributes. ‘found’ attribute is set to true or false depending on the existence of the input in the lookup table. If the input is found in the lookup table, ‘data’ attribute shows the array of all matching values. | | | | |
| Arguments: | * *table:* reference to the lookup table. * *column:* column to be looked up from the lookup table. * *index:* input attribute (string) | | | |
|  |
| Returns: | An object with two attributes:   * data: Array of matching objects * found: Boolean (True/False) | | | |
|  |
| Example: | Input type: Mailing city  lkup\_exhaustive\_m( "lkup4","city","Athens" );  Returns:   1. *{data: Array(1), found: true}*   Data.result: Array(1)  0: {zip: "4912", city: "Athens", state: "ME"}  length: 1  found: true | | | |
|  |
|  |
|  |
|  |
|  |
| **Function signature** | **lkup\_binary\_m(*table, column, index*)** | | | |
| This function is used to perform binary search on a sorted list of numeric attributes. The function returns an object with two attributes. ‘found’ attribute is set to true or false depending on the existence of the input in the lookup table. If the input is found in the lookup table, ‘data’ attribute shows the array of all matching values. | | | | |
| Arguments: | * table: reference to the lookup table. * column: column to be looked up from the lookup table. * index: input attribute (string) | | | |
| Returns: | An object with two attributes:   * data: Array of matching objects * found: Boolean (True/False) | | | |
| Example: | Input type: partial-naics  lkup\_binary\_m( "lkup17","partial\_naics","314" )   1. Return :*{data: Array(43), found: true}*   data: Array(43)  0: {partial\_naics: "314", sctg\_2digit: "18", flag\_value: "3"}  1: {partial\_naics: "314", sctg\_2digit: "19", flag\_value: "3"}  2: {partial\_naics: "314", sctg\_2digit: "20", flag\_value: "2"}  3: {partial\_naics: "314", sctg\_2digit: "21", flag\_value: "3"}  4: {partial\_naics: "314", sctg\_2digit: "22", flag\_value: "2"}  5: {partial\_naics: "314", sctg\_2digit: "23", flag\_value: "0"}  6: {partial\_naics: "314", sctg\_2digit: "24", flag\_value: "0"}  7: {partial\_naics: "314", sctg\_2digit: "25", flag\_value: "3"}  8: {partial\_naics: "314", sctg\_2digit: "26", flag\_value: "1"}  9: {partial\_naics: "314", sctg\_2digit: "27", flag\_value: "0"}  10: {partial\_naics: "314", sctg\_2digit: "28", flag\_value: "0"}  11: {partial\_naics: "314", sctg\_2digit: "29", flag\_value: "1"}  12: {partial\_naics: "314", sctg\_2digit: "30", flag\_value: "0"}  13: {partial\_naics: "314", sctg\_2digit: "31", flag\_value: "0"}  14: {partial\_naics: "314", sctg\_2digit: "32", flag\_value: "0"}  15: {partial\_naics: "314", sctg\_2digit: "33", flag\_value: "0"}  16: {partial\_naics: "314", sctg\_2digit: "34", flag\_value: "2"}  17: {partial\_naics: "314", sctg\_2digit: "35", flag\_value: "1"}  18: {partial\_naics: "314", sctg\_2digit: "36", flag\_value: "1"}  19: {partial\_naics: "314", sctg\_2digit: "37", flag\_value: "1"}  20: {partial\_naics: "314", sctg\_2digit: "38", flag\_value: "2"}  21: {partial\_naics: "314", sctg\_2digit: "39", flag\_value: "1"}  22: {partial\_naics: "314", sctg\_2digit: "40", flag\_value: "0"}  23: {partial\_naics: "314", sctg\_2digit: "41", flag\_value: "0"}  24: {partial\_naics: "314", sctg\_2digit: "43", flag\_value: "0"}  25: {partial\_naics: "314", sctg\_2digit: "99", flag\_value: "1"}  26: {partial\_naics: "314", sctg\_2digit: "17", flag\_value: "3"}  27: {partial\_naics: "314", sctg\_2digit: "16", flag\_value: "3"}  28: {partial\_naics: "314", sctg\_2digit: "15", flag\_value: "3"}  29: {partial\_naics: "314", sctg\_2digit: "14", flag\_value: "3"}  30: {partial\_naics: "314", sctg\_2digit: "13", flag\_value: "3"}  31: {partial\_naics: "314", sctg\_2digit: "12", flag\_value: "3"}  32: {partial\_naics: "314", sctg\_2digit: "11", flag\_value: "3"}  33: {partial\_naics: "314", sctg\_2digit: "10", flag\_value: "3"}  34: {partial\_naics: "314", sctg\_2digit: "09", flag\_value: "3"}  35: {partial\_naics: "314", sctg\_2digit: "08", flag\_value: "3"}  36: {partial\_naics: "314", sctg\_2digit: "07", flag\_value: "2"}  37: {partial\_naics: "314", sctg\_2digit: "06", flag\_value: "2"}  38: {partial\_naics: "314", sctg\_2digit: "05", flag\_value: "3"}  39: {partial\_naics: "314", sctg\_2digit: "04", flag\_value: "3"}  40: {partial\_naics: "314", sctg\_2digit: "03", flag\_value: "2"}  41: {partial\_naics: "314", sctg\_2digit: "02", flag\_value: "3"}  42: {partial\_naics: "314", sctg\_2digit: "01", flag\_value: "3"}  length: 43  found: true | | | |
| **Function signature** | | | **lkup\_ linear(*table, input*)** |
| This function is used search in lookup tables with one column (arrays). It returns true, if the value is found in the table, otherwise it returns false. | | | |
| Arguments: | | * *table:* reference to the lookup table. * *input: input attribute* | |
| Returns: | | Boolean (True/false) | |
| Example: | | Input type: SCTG code  lkup\_linear("lkup24","10")  Returns: True | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **check\_char(*table, input*)** |
| This function is used to check if the input string contains any of the values (substring) in the lookup table. If the input string contains any of the values in the table, it returns true, otherwise it returns false. | | |
| Arguments: | * *table:* reference to the lookup table. * *input:* input attribute | |
| Returns: | Boolean (True/false) | |
| Example: | Input type: shipment mode  check\_char("lkup26", "1")  Returns: False | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **auto\_fill(*list,attrib*)** |
| This function is used to detect the unintentional use of excel autofill function. If three consecutive values incrementing by one for a given attribute is found, it returns false. Otherwise, it returns true. | | |
| Arguments: | * *list: An array containing the entire input data* * *attribute:* The name of the variable for which autofill check is being performed | |
| Returns: | Boolean (True/False) | |
| Example: | auto\_fill("{1,2,3,4,5,6}","SHIPMENT\_VALUE");  Returns: false | |