|  |  |  |
| --- | --- | --- |
| **Function signature** | | **verify\_est**(*input*) |
| This is the main function that runs all validation functions to validate establishment’s data. It returns an array of objects containing all errors (if any) associated with the input. The objects in the returned array are arrays that show the errors associated with the input and their type, and where they happened. | | |
| Arguments: | *Input:* an array of objects each representing one line of input data | |
| Returns: | [Object{flag:String, flagmsg: String, flagname: String, flagval: Integer, line: Integer, priority: Integer}] | |
| Example: | Input: verify\_est({Mailing\_address: "Putnam Sq, no.268", Mailing\_attention: "", Mailing\_city: "Amherest", Mailing\_comapny\_name\_1: "apple34/kj", Mailing\_comapny\_name\_2: "", Mailing\_state: "MA", Mailing\_zip4: "", Mailing\_zip5: "01001", est\_id: "1", operating\_status: "Inactive", shipment\_number: "126899", shipment\_value: "66333", shipping\_address: "Mill street ave, po\_b 25668, num 23.", shipping\_city: "agawama", shipping\_comapny\_name\_1: "Oil gass 567", shipping\_comapny\_name\_2: "Patburg", shipping\_state: "DC", shipping\_zip4: "", shipping\_zip5: "01005", status\_ceased\_day: "15", status\_ceased\_month: "12", status\_ceased\_year: "2017"}]  Returns: [{flag: "E23", flagmsg: "Shipping address contains invalid characters (a value of P\_O\_/P.O\_/P..B/P.O\_/P.O./PO\_B/PO\_D/POB\_/POST).", flagname: "invChar\_shipAddress", flagval: 5, line: 1, priority: 1}, {flag: "E26", flagmsg: "City/state/zip combination is invalid.", flagname: "inv\_shipCity\_state\_zip", flagval: "23", line: 1, priority: 2}, {flag: "E27", flagmsg: "Zip code 4 digit extension is missing.", flagname: "miss\_shipZip", flagval: 2, line: 1, priority: 3}, {flag: "E30", flagmsg: "Mailing company name 2 is missing.", flagname: "miss\_companyMailName2", flagval: 2, line: 1, priority: 3}, {flag: "E30", flagmsg: "Mailing company name 2 is missing.", flagname: "miss\_companyMailName2", flagval: 2, line: 1, priority: 3}, {flag: "E35", flagmsg: "City/state/zip combination is invalid.", flagname: "inv\_mailCity\_state\_zip", flagval: "23", line: 1, priority: 3},{flag: "E36", flagmsg: "Mailing address 4 digit zip code extension is not a 5-digit number.", flagname: "inv\_mailZip", flagval: 4, line: 1, priority: 3}] | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **shipping\_company\_name\_1** *(input,line)* |
| This function validates the shipping company name. If an input is present, it checks for allowed characters (alphanumeric input is allowed). It returns an object with two attributes, pass, and errors. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: shipping company name1(string)* * *Line:* *line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: shipping\_Company\_name\_1("ship company123", 1)  Returns: Object { pass: true }  ​​ | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **shipping\_company\_name\_2***(input,line)* |
| This function validates the shipping company name. If an input is present, it checks for allowed characters (only alphanumeric input is allowed). It returns an object with two attributes, pass and, errors. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: shipping company name2 (string)* * *Line:* *line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: shipping\_Company\_name\_2("company A", 1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **shipping\_address***(input,line)* |
| This function validates the shipping address. If an input is present, it checks for allowed characters, and invalid characters (e.g. P.O.). It returns an object with two attributes, pass, and errors. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: shipping address (string)* * *Line:* *line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: shipping\_address(" 15 south shafer st", 1)  Returns: Object { pass: true } | |
| **Function signature** | | **shipping\_city***(input,line)* |
| This function validates the shipping city name. If an input is present, it checks for allowed characters (only alphabetic input is allowed). It returns an object with three attributes, pass, valid, and errors. Pass and valid are binary attributes indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: shipping city name (string)* * *Line: line number (integer)* | |
| Returns: | Object { valid: Boolean, pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: shipping\_city("Athens",1)  Returns: Object { valid: true, pass: true } | |
| **Function signature** | | **shipping\_state***(input,line)* |
| This function validates the shipping state. If an input is present, it checks for allowed characters (only alphabetic input is allowed), and it checks for input length (only abbreviation of two letters is allowed). If no errors were found, it checks for the existence of the state in a look-up table. It returns an object with three attributes, pass, valid, and errors. Pass and valid are binary attributes indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: two letters state abbreviation (string)* * *Line:* *line number (integer)* | |
| Returns: | Object { valid: Boolean, pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: shipping\_state("OH",1)  Returns: Object {valid: true, pass: true} | |
| **Function signature** | | **shipping\_zip5***(city, state, zip, evalres,line)* |
| This function validates the 5-digit shipping zip code. If an input is present, it checks for allowed characters (required to be numeric). Then, it checks the input length (required to be 5-digit length). If no errors found, it checks the existence of the entered zip code in a lookup table. If the zip code exists, it checks if it matches with the combination of the entered city and state. It returns an object with two attributes ‘pass’ and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *city: city name (string)* * *state: two letters state abbreviation (string)* * *zip:5 digit zip code (numeric)* * *evalres: an object containing the validation results for city and state attributes(object)* * *line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input type: shipping\_zip5 ("Athens", "OH", "45701", {shipping\_city:{valid:true},shipping\_state:{valid:true}}, 1)  Returns: Object {pass: true} | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **shipping\_zip4** *(input,line)* |
| This function validates the 4-digit shipping zip code extension. If an input is present, it checks for allowed characters (required to be numeric). Then, it checks the input length (required to be 4-digit length). It returns an object with two attributes, ‘pass’ and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no errors found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: 4-digit shipping zip code extension (numeric)* * *Line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input type: shipping\_zip4 ("7704",1)  Returns: Object { pass: true } | |
| **Function signature** | | **mailing\_company\_name\_1***(input,line)* |
| This function validates the mailing company name. If an input is present, it checks for allowed characters (only alphanumeric input is allowed). It returns an object with two attributes ‘pass’, and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: mailing company name1(string)* * *Line:* *line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: mailing\_company\_name\_1("mailcomp123",1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **mailing\_company\_name\_2***(input,line)* |
| This function validates the mailing company name. If an input is present, it checks for allowed characters (only alphanumeric input is allowed). It returns an object with two attributes ‘pass’ and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: mailing company name1(string)* * *Line:* *line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: mailing\_company\_name\_2("mailcomp1234",1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **mailing\_attention** *(input,line)* |
| This function validates the mailing attention. If an input is present, it checks the allowed characters. It returns an object with two attributes ‘pass’ and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: mailing attention (string)* * *line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: mailing\_attention("moh", 1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **mailing\_address***(input,line)* |
| This function validates the shipping address. If an input is present, it checks the allowed characters (alphanumeric input is allowed). It returns an object with two attributes ‘pass’ and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: mailing address (string)* * *line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: mailing\_address("15 s shafer st",1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **mailing\_city***(input,line)* |
| This function validates mailing city name. If an input is present, it checks the allowed characters (only alphabetic input is allowed). It returns an object with three attributes ‘pass’, ‘valid’, and ‘errors’. Pass and valid are binary attributes indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: mailing city name (string)* * *line: line number (integer)* | |
| Returns: | Object { valid: Boolean, pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: mailing\_city("Athens",1)  Returns: Object { valid: true, pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **mailing\_state***(input,line)* |
| This function validates the mailing state. If an input is present, it checks the allowed characters (only an alphabetic input is allowed) and length (two letter input is allowed). If no error is found, it checks for the existence of the state in a look-up table. It returns an object with three attributes ‘pass’, ‘valid’, and ‘errors’. Pass and valid are binary attributes indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: two letters state abbreviation (string)* * *Line:* *line number (integer)* | |
| Returns: | Object { valid: Boolean, pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: mailing\_state("OH",1)  Returns: Object { valid: true, pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **mailing\_zip5***(city, state, zip, evalres ,line)* |
| This function validates the 5-digit mailing zip code. If a zip code is present, it checks for allowed characters (required to be numeric). Then, it checks the input length (required to be 5-digit length). If no errors are found, it checks the existence of the entered zip code in a lookup table. If the zip code exists, it checks if it matches the combination of city and state. It returns an object with two attributes ‘pass’ and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *city: city name (string)* * *state: two letters state abbreviation (string)* * *zip:5-digit zip code (numeric)* * *evalres: an object containing the validation results for city and state attributes(object)* * *line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: mailing\_zip5("Athens","OH", "45701", {mailing\_city:{valid:true},mailing\_state:{valid:true}}, 1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **mailing\_zip4** *(input,line)* |
| This function validates the 4-digit mailing zip code extension. If a zip code is present, it checks the allowed characters (required to be numeric). Then, it checks the input length (required to be 4-digit length). It returns an object with two attributes, ‘pass’, and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *Input: 4-digit mailing zip code extension (numeric)* * *Line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: mailing\_zip4("7704",1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **date\_Of\_Ceased***(month, day, year, line)* |
| This function validates the date of ceased operation (month, day, and year). If an input is present, it checks the allowed characters (numeric input for month, day and year are allowed). If no error is found, it checks the range for month, day, and year values. It returns an object with two attributes ‘pass’ and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Arguments: | * *month: month of ceased operation (string)* * *day: day of ceased operation (string)* * *year: year of ceased operation (string)* * *line: line number (integer)* | |
| Returns: | Object {Valid: Boolean, pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: date\_Of\_Ceased("11","28","2018",1)  Returns: Object { valid: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **Verfication\_priamaryIndustry\_checkBox** *(input,line)* |
| This function validates the input presence for primary industry check box. It returns an object with two attributes, ‘pass’, and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Argument : | * *Input: primary industry check box* * *line: line number (Integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: Verfication\_priamaryIndustry\_checkBox (‘true’,1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **contact\_name***(input,line)* |
| This function validates the contact name. If the input is present, it checks for allowed characters (alphabetic input is required). It returns an object with two attributes, ‘pass’, and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Argument : | * *Input: contact name (String)* * *line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: contact\_name("John",1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **contact\_title***(input,line)* |
| This function validates the contact title. If the input is present, it checks for allowed characters (alphabetic input is required). It returns an object with two attributes, ‘pass’, and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Argument : | * *Input: contact title (string)* * *line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input : contact\_title("Mr.",1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **contact\_phone***(input1, exten, line)* |
| This function validates the contact title (both of ‘input1’, and ‘exten’). If input1 is present, it checks for allowed characters (numeric input is required). It also checks for field length (it is required to be a 10 digits number). Next, if ‘exten’ is present, it checks for its field length (it is required to be between 1 and 7). It returns an object with two attributes, ‘pass’, and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Argument : | * *Input1: contact phone number(string)* * *Exten: phone number extension (string)* * *line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input : contact\_phone("7405910647", "123",1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **contact\_fax\_number***(input,line)* |
| This function validates the contact fax number. If input is present, it checks for allowed characters (numeric input is required). It also checks for field length (it is required to be a 10 digits number). It returns an object with two attributes, ‘pass’, and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Argument : | * *Input: contact fax number (string)* * *line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: contact\_fax\_number("7405930286",1)  Returns: Object { pass: true } | |

|  |  |  |
| --- | --- | --- |
| **Function signature** | | **completion\_time***(input1, input2, line)* |
| This function validates the completion time of the survey (hours and minutes). If input1 and input2 are present, it checks for the allowed characters for both inputs (numeric input is required). Next, it checks for the value range of the inputs (maximum allowed is 8 hours for input1). It returns an object with two attributes, ‘pass’, and ‘errors’. Pass is a binary attribute indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Argument : | * *Input1: completion time number of hours (string)* * *Input2: completion time number of minutes(string)* * *Line: line number (integer)* | |
| Returns: | Object {pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input: completion\_time("7","10",1)  Returns: Object { pass: true } | |
| **Function signature** | | **remarks***(input,line)* |
| This function validates the response provided for ‘remarks’ field. It returns an object with three attributes, ‘pass’, valid, and ‘errors’. Pass and valid are binary attributes indicating whether the input has any errors. It is set to true if no error is found, otherwise, it is set to false. Errors is an array of objects reflecting the list of detected errors. | | |
| Argument : | * *Input: remarks (string)* * *line: line number (integer)* | |
| Returns: | Object {valid: Boolean, pass: Boolean, errors: [ Object {flag: String, flagmsg: string, flagname: string, flagval: int, line: int, priority: int }]} | |
| Example: | Input type: remarks("This is a test",1)  Returns: Object { valid: true, pass: true } | |