**main package:**

line 184: the valchan is where the value is returned, okchan shows whether the operation was successful or not, also the case statement for either valchan or okchan will be run depending upon what comes first

line 187: the case statement will check if the value is empty or non empty

**package\_func:**

line 106: if key does not exist, if it doesn’t then we return the valchan as empty string and okchan as false, this is because d.data[key] returns the value and the ok variable, ok shows whether key exists or not

line 115: checks if the queue is empty if it is then we again return the same values as above

line 121: first value from the queue is retrieved and stored in value

line 122: queue is sliced to remove the first value that we received in the previous line

line 124 and line 125: we return the values via the channels

line 41: fset is used to store object is used to generate source code position information for every node in the ast

ast is generated by the parser.parseexpr

fset.position is then used to print position info along with the token using the ast.inspect call

fset object keeps track of positions of the tokens

line 42: this line will catch all key value pair by parsing and S.printf converts it into string and all of this captured in expr object (which is actually ast.expr)

line 46: ast.inspect recursively on every node which is in the expr object

**main\_test**

line 49: We are creating a slice of index values at the index as [1,2,3,4]

line 55: we reset the timer to start testing the time of pushing values