```
#########################################################################################################################

import arcpy
import os
import sys
sys.path.append(r'E:\GIS 5253-GIS Applications')
import belm0008_module

# Setting workspace environment
arcpy.env.workspace = r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Thematic Mapping GIS 5253.gdb"

# List of Counties of 2018 and their 2018 Median Income tables to be joined via Block Group
counties18 = [
   {"shapefile": "Cameron18", "table": r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Incomes\5Y2018.MedInc.Cameron.xls\T_ACSDT5Y2018_B19013_Data$_"},
   {"shapefile": "Hildalgo18", "table": r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Incomes\5Y2018.MedInc.Hildalgo.xls\T_ACSDT5Y2018_B19013_Data$_"},
   {"shapefile": "Starr18", "table": r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Incomes\5Y2018.MedInc.Starr.xls\T_ACSDT5Y2018_B19013_Data$_"},
   {"shapefile": "Willacy18", "table": r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Incomes\5Y2018.MedInc.Willacy.xls\T_ACSDT5Y2018_B19013_Data$_"},]

# List of Counties of 2022 and their 2022 Median Income tables to be joined via Block Group
counties22 = [
   {"shapefile": "Cameron22", "table": r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Incomes\5Y2022.MedInc.Cameron.xls\T_ACSDT5Y2022_B19013_Data$_"},
   {"shapefile": "Hildalgo22", "table": r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Incomes\5Y2022.MedInc.Hildalgo.xls\T_ACSDT5Y2022_B19013_Data$_"},
   {"shapefile": "Starr22", "table": r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Incomes\5Y2022.MedInc.Starr.xls\T_ACSDT5Y2022_B19013_Data$_"},
   {"shapefile": "Willacy22", "table": r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Incomes\5Y2022.MedInc.Willacy.xls\T_ACSDT5Y2022_B19013_Data$_"},]

# Join field names in table and county shapefile
join_shapefile = "GEOID"
join_table = "BGroupID"

# Please see module code
belm0008_module.process_export
```

```
# Check if the shapefile exists in the workspace
for county_shapefile in county_medinc18:
  shapefile_path = os.path.join(workspace, county_shapefile)
  if arcpy.Exists(shapefile_path):
    existing_shapefiles.append(shapefile_path)
  else:
    print(f"Shapefile {county_shapefile} does not exist in the workspace.")

# If valid shapefiles are found, the merge will be performed
if existing_shapefiles:
  arcpy.management.Merge(
    inputs=";".join(existing_shapefiles),
    output=os.path.join(workspace, "RGV18_MedianIncome"), # Taken from the Geoprocessing python window
    field_mappings=None,
    add_source="NO_SOURCE_INFO",
    field_match_mode="AUTOMATIC")
  print(f"Merged shapefiles: {existing_shapefiles}")
else:
  print("No valid shapefiles found to merge.")



# Now the same process is performed to merge the counties into Rio Grande Valley 2022 (RGV)

county_medinc22 = ["Willacy22_MedianIncome", "Starr22_MedianIncome", "Hildalgo22_MedianIncome", "Cameron22_MedianIncome"]

# Establish an empty list to hold the existing shapefiles
existing_shapefiles1 = []

# Check if the shapefile exists in the workspace
for county_shapefile1 in county_medinc22:
  shapefile_path1 = os.path.join(workspace, county_shapefile1)
  if arcpy.Exists(shapefile_path):
    existing_shapefiles1.append(shapefile_path1)
  else:
    print(f"Shapefile {county_shapefile1} does not exist in the workspace.")

# If valid shapefiles are found, the merge will be performed
if existing_shapefiles1:
  arcpy.management.Merge(
    inputs=";".join(existing_shapefiles1),
    output=os.path.join(workspace, "RGV22_MedianIncome"), # Taken from the Geoprocessing python window
    field_mappings=None,
    add_source="NO_SOURCE_INFO",
    field_match_mode="AUTOMATIC")
  print(f"Merged shapefiles: {existing_shapefiles1}")
else:
  print("No valid shapefiles found to merge.")
```

```
# Looking at county  tabular data in the RGV 2018 and 2022 Shapfiles

# Defining new variables to the created RGV MedianIncome Files
RGV22 = r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Thematic Mapping GIS 5253.gdb\RGV18_MedianIncome"
RGV18 = r"E:\GIS 5253-GIS Applications\Thematic Mapping GIS 5253\Thematic Mapping GIS 5253.gdb\RGV22_MedianIncome"

#Requesting user input regarding COUNTYFP code to show cause rows from attribute table
countyfp_code = input("Enter the COUNTYFP code from the following (Cameron 061, Hildalgo 215, Willacy 489, Starr 427): ")

# Define a function to search for a COUNTYFP in a shapefile and display the row
def row_in_rgv(shapefile, countyfp_code):
    rows_found = []
    with arcpy.da.SearchCursor(shapefile, ["COUNTYFP", "GEOID", "NAMELSAD", "B19013_001E"]) as cursor:
        for row in cursor:
            if row[0] == countyfp_code:
                rows_found.append(row)
    return rows_found

# Search for the COUNTYFP in both shapefiles
Row_RGV22 = row_in_rgv(RGV22, countyfp_code)
Row_RGV18 = row_in_rgv(RGV18, countyfp_code)

print(arcpy.ListFields(RGV22))
print(arcpy.ListFields(RGV18))

# Display results
if Row_RGV22:
    print(f"Results from RGV22 shapefile for COUNTYFP {countyfp_code}:")
    for row in Row_RGV22:
        print(f"GEOID: {row[1]}, Name: {row[2]}, B19013_001E: {row[3]}")
else:
    print(f"No results found in RGV22 for COUNTYFP {countyfp_code}.")

if Row_RGV18:
    print(f"Results from RGV18 shapefile for COUNTYFP {countyfp_code}:")
    for row in Row_RGV18:
        print(f"GEOID: {row[1]}, Name: {row[2]}, B19013_001E: {row[3]}")
else:
    print(f"No results found in RGV18 for COUNTYFP {countyfp_code}.")
```

```python
# Defining exports to process joins and export shapefiles with join features
def process_export(counties):
    for county in counties:
        shapefile = county["shapefile"]
        table = county["table"]
        try:
            # Joining shapefiles to table
            arcpy.management.AddJoin(shapefile, join_shapefile, table, join_table)
            print(f"Successfully joined table {table} to shapefile {shapefile}.")

            # Export the joined shapefile directly to the geodatabase
            out_name = f"{shapefile}_MedianIncome"

            # Export the joined feature class
            arcpy.conversion.FeatureClassToFeatureClass(shapefile, arcpy.env.workspace, out_name)
            print(f"Feature '{shapefile}' exported as '{out_name}' in the geodatabase.")
        except Exception as e:
            print(f"Failed to process shapefile {shapefile} with table {table}. Error: {e}")
```