var geometry = ee.FeatureCollection("users/saivilliers/Iraq");

Map.centerObject(geometry, 7);var scaleMOD = function(img) {  
        return  img  
          .divide(10000)  
          .float()  
          .set("system:time\_start", img.get("system:time\_start"));  
};var modis = ee.ImageCollection("MODIS/006/MOD13Q1")  
            .map(scaleMOD)  
            .filterBounds(geometry)  
            .select('NDVI');var startyear = 2000;  
var endyear = 2021;  
print ('no of years in study period', endyear - startyear + 1);  
var startmonth = 4;  
var endmonth = 5;  
print ('no of months in study period', endmonth - startmonth + 1);  
var startdate = ee.Date.fromYMD(startyear, startmonth, 1);  
var enddate = ee.Date.fromYMD(endyear , endmonth, 30);  
var years = ee.List.sequence(startyear, endyear);  
var months = ee.List.sequence(startmonth,endmonth);var modis\_list = [];  
for(var i = startmonth; i <=endmonth; i++) {  
  for(var j = startyear; j <=endyear; j++){  
    modis\_list.push(modis.filter(ee.Filter.calendarRange(i, i, 'month'))  
                          .filter(ee.Filter.calendarRange(j, j, 'year')));  
  }  
}  
print(modis\_list);  
var modis\_filtered\_flattened = ee.ImageCollection(ee.FeatureCollection(modis\_list).flatten());// NDVI composites for every month  
var monthlyNDVI =  ee.ImageCollection.fromImages(  
  years.map(function (y) {  
    return months.map(function(m) {  
      var monthly = modis\_filtered\_flattened  
        .filter(ee.Filter.calendarRange(y, y, "year"))  
        .filter(ee.Filter.calendarRange(m, m, "month"))  
        .mean();  
      return monthly  
        .set("year", y)  
        .set("month", m)  
        .set("system:time\_start", ee.Date.fromYMD(y, m, 1).millis());}); })  
  .flatten());  
print (monthlyNDVI, 'monthly NDVI composites in study period = no of months \* no of years')  
Map.addLayer (monthlyNDVI.first().clip(geometry),  {min:0, max:1,  'palette': ['red','yellow', 'green']}, 'MonthlyNDVI')// Extract max values for each month over all years  
var MonthlyMAX =  ee.ImageCollection.fromImages(months  
  .map(function (m) {  
    var maxNDVI = monthlyNDVI  
      .filter(ee.Filter.eq("month", m))  
      .select("NDVI")  
      .reduce(ee.Reducer.max())  
      .rename("max\_NDVI");  
  return maxNDVI  
    .set("month", m);})  
  .flatten());  
print (MonthlyMAX, 'MonthlyMAX no of images = no of months in study period');  
Map.addLayer (MonthlyMAX.first().select('max\_NDVI').clip(geometry),  {min:0, max:1,  'palette': ['red','yellow', 'green']}, 'MonthlyMAX')// Extract min values for each month over all years  
var MonthlyMIN =  ee.ImageCollection.fromImages(months  
  .map(function (m) {  
    var minNDVI = monthlyNDVI  
      .filter(ee.Filter.eq("month", m))  
      .select("NDVI")  
      .reduce(ee.Reducer.min())  
      .rename("min\_NDVI");  
  return minNDVI  
    .set("month", m);})  
  .flatten());  
print (MonthlyMIN, 'MonthlyMIN  no of images = no of months in study period');  
Map.addLayer (MonthlyMIN.first().select('min\_NDVI').clip(geometry), {min:0, max:1,  'palette': ['red','yellow', 'green']}, 'MonthlyMIN')// Extract VCI for each month over all years  
var vci\_monthly = monthlyNDVI.map(function(img){  
 var id = img.id();  
 var min =  img.reduceRegion(ee.Reducer.min(), geometry,250).get('NDVI');  
 var max = img.reduceRegion(ee.Reducer.max(), geometry,250).get('NDVI');  
 return img.expression(  
   "((NDVI-min)/(max-min))\*100",{  
     "NDVI" : img,  
     "max" : ee.Number(max),  
     "min" : ee.Number(min)  
   }).copyProperties(img,['system:time\_start','system:time\_end']);  
});print(vci\_monthly)  
print (vci\_monthly, 'vci\_monthly composites in study period = no of months \* no of years')  
Map.addLayer (vci\_monthly.first().clip(geometry),  {min:-100, max:100,  'palette': ['red','yellow', 'green']}, 'vci\_monthly')