```powershell
<#
  .SYNOPSIS
    This file makes a backup of all proxy addresses of AzureAD users in Office 365.

  .PARAMETER AllOutputFile
    This file records the information for all users in the WVUHSC tenant

  .PARAMETER HSCOutputFile
    This file only records information for hsc users in the WVUHSC tenant

  .NOTES
    Written by: Jeff Brusoe
    Last Updated: November 19, 2020
#>

[CmdletBinding()]
param(
  [ValidateNotNullOrEmpty()]
  [string]$AllOutputFile = ("$PSScriptRoot\Logs\" +
              (Get-Date -Format yyyy-MM-dd-hh-mm) +
              "-O365ProxyAddresses.csv"),

  [ValidateNotNullOrEmpty()]
  [string]$HSCOutputFile = ("$PSScriptRoot\Logs\" +
              (Get-Date -Format yyyy-MM-dd-hh-mm) +
              "-HSCO365ProxyAddresses.csv")
)

#Initialize environment
try {
  Set-HSCEnvironment -ErrorAction Stop

  $Office365 = Connect-HSCOffice365 -ErrorAction Stop
  Write-Output "Office 365 Connection Status: $Office365"
}
catch {
  Write-Warning "Error configuring environment"
  Invoke-HSCExitCommand -ErrorCount $Error.Count
}

Write-Output "Output File: $AllOutputFile"
Write-Output "HSC Outtput File: $HSCOutputFile"

#Begin main part of program
Write-Output "Generating list of AzureAD Users"
try {
  $AzureADUsers = Get-AzureADUser -All $true -ErrorAction Stop
}
catch {
  Write-Warning "Unable to generate list of AzureAD users"
  Invoke-HSCExitCommand -ErrorCount $Error.Count
}

foreach ($AzureADUser in $AzureADUsers)
{
  Write-Output $("Current User: " + $AzureADUser.UserPrincipalName)

  $ProxyAddresses = $AzureADuser.ProxyAddresses #| Where-Object {$_ -match "^smtp"}
  Write-Output "ProxyAddresses:"
  Write-Output $ProxyAddresses
```

```powershell

  $Licensed = $false
  $LicenseCount = ($AzureADUser.AssignedLicenses | Measure-Object).Count
  if ($LicenseCount -gt 0) {
    $Licensed=$true
  }

  $AzureADUser |
    Select-Object UserPrincipalName,
            mail,
            @{name="ProxyAddresses";expression={$ProxyAddresses -join ";"}},
            @{name="UserLicensed";expression={$Licensed}} |
    Export-Csv $AllOutputFile -NoTypeInformation -Append

  #This code was put here at the request of WVUM to generate a file that they wanted.
  if ($AzureADUser.UserPrincipalName -like "*hsc.wvu.edu*") {
    Write-Output "HSC AD User - Getting AD Info for WVUM"

    $LDAPFilter = "(userPrincipalName=" + $AzureADUser.UserPrincipalName + ")"

    $GetADUserParams = @{
      LDAPFilter = $LDAPFilter
      Properties = @(
        "mail",
        "extensionAttribute11",
        "extensionAttribute13"
      )
      ErrorAction = "Stop"
    }

    try {
      $ADUser = Get-ADUser @GetADUserParams
      Write-Output "Successfully found AD User"
    }
    catch {
      Write-Warning "Error searching for AD User"
    }

    if (![string]::IsNullOrEmpty($ADUser.mail)) {
      $ADmail = $ADUser.mail
    }
    else {
      $ADmail = "None"
    }

    if (![string]::IsNullOrEmpty($ADUser.extensionAttribute11)) {
      $WVUID = $ADUser.extensionAttribute11
    }
    else {
      $WVUID = "None"
    }

    if (![string]::IsNullOrEmpty($ADUser.extensionAttribute13)) {
      $WVUMID = $ADUser.extensionAttribute13
    }
    else {
      $WVUMID = "None"
    }

```

```powershell
    $HSCUserinfo = [PSCustomObject]@{
      SamAccountName = $ADUser.SamAccountName
      UserPrincipalName = $AzureADUser.UserPrincipalName
      ADMailAttribute = $ADMail
      WVUID = $WVUID
      WVUMID = $WVUMID
      UserLicensed = $Licensed
      ProxyAddresses = ($ProxyAddresses -join ";")
    }

    $HSCUserInfo | Export-Csv $HSCOutputFile -NoTypeInformation -Append

  }

  Write-Output "***********************************"
}

Invoke-HSCExitCommand -ErrorCount $Error.Count
```