```
function BennaPrecipitationFilename, orbit, mnum, proton=proton, $
  electron=electron, params=params

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; mnum =
;;   0: inbound IMF conditions, Best fit
;;   1: outbound IMF conditions, Best fit
;;   2: inbound IMF conditions, low density
;;   3: inbound IMF conditions, medium density
;;   4: inbound IMF conditions, high density
;;   5: outbound IMF conditions, low density
;;   6: outbound IMF conditions, medium density
;;   7: outbound IMF conditions, high density
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

if (proton EQ !null) then proton = 0
if (electron EQ !null) then electron = 0
if (proton+electron EQ 0) then proton = 1
if (proton+electron NE 1) then stop

if (mnum EQ !null) then begin
  print, '  0 = Inbound IMF, Best fit'
  print, '  1 = Outbound IMF, Best fit'
  print, '  2 = Inbound IMF, low density SW'
  print, '  3 = Inbound IMF, medium density SW'
  print, '  4 = Inbound IMF, high density SW'
  print, '  5 = Outbound IMF, low density SW'
  print, '  6 = Outbound IMF, medium density SW'
  print, '  7 = Outbound IMF, high density SW'
  read, mnum, prompt='Enter the IMF conditions: '
endif
if ((mnum LT 0) or (mnum GT 7)) then stop

case (mnum) of
  0: begin
     imfstr = 'Inbound IMF, best fit'
     o = 0
     end
  1: begin
     imfstr = 'Outbound IMF, best fit'
     o = 1
     end
  2: begin
     imfstr = 'Inbound IMF, low density'
     o = 0
     end
  3: begin
     imfstr = 'Inbound IMF, medium density'
     o = 0
     end
  4: begin
     imfstr = 'Inbound IMF, high density'
     o = 0
     end
  5: begin
     imfstr = 'Outbound IMF, low density'
     o = 1
     end
  6: begin
     imfstr = 'Outbound IMF, medium density'
     o = 1
     end
  7: begin
     imfstr = 'Outbound IMF, high density'
     o = 1
     end
endcase

;; Determine which model to use
restore, !model.basepath + 'Work/Data/surfacemaps/Mercury/PrecipModelCrossRef.sav'

q = (where(*precip_orbit.orbit EQ orbit, nq))[0]
if (nq NE 1) then stop

modelnumber = (*precip_orbit.models)[mnum,q]

if (modelnumber NE -1) then begin
  modelden = (*precip_orbit.mod_den)[mnum,q]
  modelBx = (*precip_orbit.mod_Bx)[mnum,q]
  modelBy = (*precip_orbit.mod_By)[mnum,q]
  modelBz = (*precip_orbit.mod_Bz)[mnum,q]

  Bx = (*precip_orbit.Bx)[o,q]
  By = (*precip_orbit.By)[o,q]
  Bz = (*precip_orbit.Bz)[o,q]

  ;; Determine name of precipitation file
  case (1) of
    (modelnumber LT 10): mstr = '000' + strint(modelnumber)
    (modelnumber LT 100): mstr = '00' + strint(modelnumber)
    (modelnumber LT 1000): mstr = '0' + strint(modelnumber)
    else: mstr = strint(modelnumber)
  endcase

  part = (proton) ? 'Proton' : 'Electron'
  filename = !model.basepath + 'Work/Data/surfacemaps/Mercury/' + $
    part + 'Precipitation/' + mstr + '.' + part + '.sav'

  params = {orbit:orbit, IMF:imfstr, model:modelnumber, filename:filename, $
    modelden:modelden, modelbx:modelbx, modelby:modelby, modelbz:modelbz, $
    bx:bx, by:by, bz:bz}

  print, 'Orbit # = ' + strint(orbit)
  print, 'Model # = ' + strint(modelnumber)
  print, 'IMF conditions: ' + imfstr
  print, 'Model Density = ' + strint(modelden)
  print, 'Bx: Observed = ' + strint(Bx) + ' Modeled = ' + strint(modelbx)
  print, 'By: Observed = ' + strint(By) + ' Modeled = ' + strint(modelby)
  print, 'Bz: Observed = ' + strint(Bz) + ' Modeled = ' + strint(modelbz)
endif else begin
  filename = ''
  print, 'Orbit # = ' + strint(orbit)
  print, 'IMF conditions: ' + imfstr
  print, 'No model satisfies these conditions'
endelse

return, filename

end
```