


```

pro EXPAND,a,nx,ny,result,maxval=maxval,fillval=fillval

s=size(a)
if(s(0) ne 2) then begin
  print,'EXPAND: *** array must be 2-Dimensional ***'
  retall ; This will completely terminate the MAIN program!!!
endif

; Get dimensions of the input array
ix = s(1)
iy = s(2)

; Calculate the new grid in terms of the old grid
ux = (ix-1.) * findgen(nx) / (nx-1.)
uy = (iy-1.) * findgen(ny) / (ny-1.)

; Are we to look for and ignore bad data? (can't use KEYWORD_SET here)
if(n_elements(maxval) eq 0) then begin
; NO
  result = interpolate(a,ux,uy,/grid)
endif else begin
; YES then calculate the indicies and u-arrays
  mx = long(ux)<(ix-2)
  my = long(uy)<(iy-2)
  uxa = ux # replicate(1,ny)
  uya = replicate(1,nx) # uy

;Index vectors to A and RESULT arrays
  mxy = (mx # replicate(1L,ny)) + (replicate(long(ix),nx) # my)
  ind = lindgen(nx,ny)

; Fill RESULT with fill value, defaulting to -1 if none specified
if(n_elements(fillval) le 0) then fillval = -1.
result = replicate(fillval,nx,ny)

; Remove those elements which would be utilizing "bad" values from A
; Check lower left
  m = where(a(mxy) le maxval,num)
  if(num eq 0) then goto,out
  mxy = mxy(m)
  ind = ind(m)
; Check lower right
  m = where(a(mxy+1) le maxval,num)
  mxy = mxy(m)
  ind = ind(m)
; Check upper left
  m = where(a(mxy+ix) le maxval,num)
  mxy = mxy(m)

```

```
ind  = ind(m)
; Check upper right
m    = where(a(mxy+(ix+1)) le maxval,num)
mxy  = mxy(m)
ind  = ind(m)

; Interpolate only the points which will not be the fill value
result(ind) = interpolate(a,uxa(ind),uya(ind))
endelse

; Done
return
OUT: ; If we had a problem
print,'Entire input array is greater than MAXVAL, ('+strtrim(maxval,2)+')'
return
end
```
