OpenMP exercises (F.Vana)

Following examples are trying to highlight typical race problems appearing
in the code.

* Exercise No 1 (using arp/adiab/cpg.90 code)
Add: i/ one dummy array ZDUMMT1 of size at least (NPROMA,NFLEVG) to pass from
     CPG_GP to and MF_PHYS
   ii/ one dummy array ZDUMMT2 of size at least (NPROMA,NFLEVG) to pass from
     MF_PHYS_PREP and MF_PHYS
   The aim is indeed to respect OpenMP (for both options of LDCPG_SPLIT) and
   to declare the fields ZDUMMT1 and ZDUMMT2 with the minimal dimensions.
Hints:
1/ Try to understand the difference between arrays like ZDELP0, ZMOCON and GFL.
2/ Try to understand functionality of the LDCPG_SPLIT key

* Exercise No 2 (using arp/phys_dmn/aplpar.F90, actqsats.F90 and acmodo.F90 ...)

(Warning: Consider this exercise as a very abstract one. Don't try to find
     any physical meaning there.)

In the downdraught part of 3MT we would like to redefine the YOMCST0 constant
RTT with respect to surface temperature (PTS). The value of RTT defining the
temperature of water fusion at normal pressure) should be then restored once
the code under LCDDPRO key in APLPAR is left.

It basically means to add following three lines (labeled as "<---")
under APLPAR (plus the declaration of local variable ZORIGINAL_RTT):

```
  IF (LCDDPRO) THEN
  ! ------------------------
  ! DOWNDRAUGHT CONTRIBUTION
  ! ------------------------

  ZORIGINAL_RTT=RTT                      <---
  RTT=MIN(MINVAL(PTS(KIDIA:KFDIA)),263.16_JPRB)  <---

  ...

  RTT=ZORIGINAL_RTT                      <---

  ENDIF ! LCDDPRO
```


This however is not threadsafe, as all APLPAR computation is under

dynamic extent of OpenMP (parallel region).

Try to propose a solution solving this example of redefining
shared variable within a parallel region.

Hints: Note, that ACTQSATS routine is used also outside
       the LCDDPRO areas.
       There are always more solutions. If you like, you can
       propose several alternatives.

* Exercise No 3 (using arp/adiab/cpg.F90 and ???)

Optimization example from the real code.

Prior to the MF physics (MF_PHYS), there's routine MF_PHYS_PREP precomputing
some arrays required mostly by AROME physics. The transfer of those
quantities between the MF_PHYS_PREP and MF_PHYS is realized through
SHARED arrays. This is absolutely unnecessary there's a place
for optimization by replacing the relevant fields by PRIVATE arrays.
In such case especially with LDCPG_SPLIT=.TRUE. key the model running
with OpenMP will require less memory.

Do the cleaning/optimization as phasing exercise.