```
# [...]

# SOLUTION 2017-01-18

#-----------------------------------------------------------------------------
# Load IDT entries.
#-----------------------------------------------------------------------------
    carica_gate TIPO_R          a_reg            LIV_UTENTE
        carica_gate     TIPO_LS         a_listen         LIV_UTENTE
        carica_gate     TIPO_B          a_broadcast      LIV_UTENTE

# SOLUTION 2017-01-18

# [...]

# SOLUTION 2017-01-18

#-----------------------------------------------------------------------------
# IDT entries subroutines definitions.
#-----------------------------------------------------------------------------
# Registers the calling process as either a listener or a broadcaster. One of
# these roles must be specified and one and only one process can be registered
# as broadcaster.
#-----------------------------------------------------------------------------
.GLOBAL a_reg
#-----------------------------------------------------------------------------
a_reg:
    .cfi_startproc
    .cfi_def_cfa_offset 40
    .cfi_offset rip, -40
    .cfi_offset rsp, -16
    call c_reg
    iretq
    .cfi_endproc

#-----------------------------------------------------------------------------
.GLOBAL a_listen
#-----------------------------------------------------------------------------
# The listen() primitive will hang the calling process if all messages have
# already been delivered until the next broadcast message is sent. At the of the
# C++ implementation c_listen() the calling process is placed in the global
# broadcast descriptor listeners queue if its b_id (last retrieved broadcast
# message id) is equal to the system broadcast last_id and the scheduler
# is called. This is why we have to save the current process state (salva_stato)
# and load a new process (carica_stato).
#-----------------------------------------------------------------------------
a_listen:
    .cfi_startproc
    .cfi_def_cfa_offset 40
    .cfi_offset rip, -40
    .cfi_offset rsp, -16
    call salva_stato
    call c_listen
    call carica_stato
    iretq
    .cfi_endproc

#-----------------------------------------------------------------------------
.GLOBAL a_broadcast
#-----------------------------------------------------------------------------
# The broadcast() primitive will move the calling process to the system ready
# processes queue after delivering the broadcast message to the available
# listeners. At the end of the C++ implementation a new process is scheduled.
# That’s why we need to save the current process (broadcaster) process and load
# a new process state (the scheduler is called at the end of the C++
# implementation).
#-----------------------------------------------------------------------------
a_broadcast:
    .cfi_startproc
    .cfi_def_cfa_offset 40
```

```
    .cfi_offset rip, -40
    .cfi_offset rsp, -16
    call salva_stato
    call c_broadcast
    call carica_stato
    iretq
    .cfi_endproc

# SOLUTION 2016-09-20
```