```
# ******************************************************************************
# Arquitectura de Computadores I
# Ano lectivo 2011/12
#
# Correcção do teste prático 2 (13/01/2012)
#

# ******************************************************************************
# Questão 1
# ******************************************************************************
#
# Mapa de Registos:
#   n:   $a0 -> $s0
#   i:   $s1
#   sum: $f20

        .data
um:     .double 1.0
dois:   .double 2.0

        .text
        .globl  eval

eval:   subu    $sp,$sp,20
        sw      $ra, 0($sp)
        sw      $s0, 4($sp)
        sw      $s1, 8($sp)
        s.d     $f20, 12($sp)

        move    $s0,$a0
if:     bne     $s0,$0,else     # if (n == 0)
        l.d     $f0,um          #   return 1.0
        j       final           #
else:                           # else
                                # {
        sub.d   $f20,$f20,$f20  #   sum =0.0
                                #
        li      $s1,0           #   i = 0
for:    bge     $s1,$s0,endfor  #   for(; i < n; i++)
                                #   {
        move    $a0,$s1         #
        jal     eval            #
        add.d   $f20,$f20,$f0   #     sum += eval(i);
        addi    $s1,$s1,1       #     i++
        j       for             #   }
endfor: l.d     $f0,dois        #
        mul.d   $f0,$f0,$f20    #   $f0 = 2.0 * sum
        mtc1    $s0,$f2         #
        cvt.d.w $f2,$f2         #
        div.d   $f0,$f0,$f2     #   $f0 = 2.0 * sum / n
        add.d   $f0,$f0,$f2     #   return (2.0 * sum / n) + n;
final:                          # }
        lw      $ra, 0($sp)
        lw      $s0, 4($sp)
        lw      $s1, 8($sp)
        l.d     $f20, 12($sp)
        addu    $sp,$sp,20
        jr      $ra
```

```
# ******************************************************************************
# Questão 2
# ******************************************************************************
#

# Estrutura             Offset
# typedef struct
# {
#   char  name[18];     0
#   float price;        20
#   char  flag;         24
#   int   qtd;          28
# } BOOK;
#
# sizeof(BOOK) = 32

# Mapa de Registos:
#   nelem:  $s0
#   status: $v0
#   value:  $f20
#   argc:   $a0
#   argv:   $a1
#
        .data
txt:    .asciiz "Pesquisa com sucesso"
        .align  2
book_array:
        .space  320
max:    .space  4

        .text
        .globl  main
main:   subu    $sp,$sp,12
        sw      $ra, 0($sp)
        sw      $s0, 4($sp)
        s.s     $f20,8($sp)
if:     bge     $a0,1 endif     # if (argc < 1)
        li      $v0,-1          #   return -1
        j       final           #
endif:  lw      $a0,0($a1)      #
        jal     atoi            #
        move    $s0,$v0         # nelem = atoi(argv[1]);
        li      $v0,6           #
        syscall                 #
        mov.s   $f20,$f0        # value = read_float();
        la      $a0, book_array #
        move    $a1, $s0        #
        jal     read_array      # read_array(book_array, nelem);
        la      $a0, book_array #
        move    $a1, $s0        #
        mov.s   $f12, $f20      #
        la      $a2, max        #
        jal     exists          # status = exists(book_array, nelem, value, &max);
        bne     $v0, 1, endif2  # if (status == 1)
        la      $a0, txt        #
        li      $v0, 4          #
        syscall                 #    print_str("Pesquisa com sucesso")
endif2: li      $v0,0           # return 0;
final:  lw      $ra, 0($sp)
        lw      $s0, 4($sp)
        l.s     $f20,8($sp)
        addu    $sp,$sp,12
        jr      $ra
```

```
# ********************************************************************************
# Questão 3
# ********************************************************************************
#
# Mapa de Registos:
#   array:  $a0
#   nelem:  $a1
#   val:    $f12
#   max:    $a2
#   i:      $t0
#   index:  $t1
#   mx:     $f2
#
        .text
        .globl  exists

exists: li      $t0,0           # i=0;
        li      $t1,-1          # index = -1;
        sub.s   $f2,$f2,$f2     # mx = 0.0;
while:  bge     $t0, $a1, endwhile # while(i < nelem)
                                # {
        sll     $t3, $t0, 5     #
        addu    $t3,$a0, $t3    #   $t3 = &array[i]
        l.s     $f4,20($t3)     #   $f4 = array[i].price
if:     c.le.s  $f4,$f2         #
        bc1t    end_if          #
        c.le.s  $f4,$f12        #
        bc1f    end_if          #   if((array[i].price>mx) && (array[i].price<=val))
                                #   {
        move    $t1,$t0         #     index = i;
        mov.s   $f2,$f4         #     mx = array[i].price
                                #   }
endif:  addiu   $t0,$t0,1       #   i++
        j       while           # }
                                #
endwhile:                       #
        beq     $t1,-1,else     # if (index != -1)
                                # {
        sll     $t4, $t1, 5     #
        addu    $t4,$a0,$t4     #   $t4 = &array[index]
        li      $t3,1           #
        sb      $t3, 24($t4)    #   array[index].flag = 1;
        s.s     $f2, 0($a2)     #   *max = mx;
        li      $v0, 1          #   return 1;
        j       final           # }
else:   li      $v0, 0          # return 0;
final:  jr      $ra


# ********************************************************************************
```

```
# Questão 4
# ********************************************************************************
#
# Mapa de resgistos:
#   array:   $a0
#   nelem:   $a1
#   i:       $t0
#   sum:     $f0
#   ptarray: $t1

        .text
        .globl  total

total:  li      $t0,1           # i=1
        sub.s   $f0,$f0,$f0     # sum = 0.0
        mov.s   $f4,$f0         # aux = 0.0
        move    $t1,$a0         # ptarray = array
        sll     $t2,$a1,3       #
        addu    $t2,$t2,$a0     # $t2 = array + nelem
for:    bgeu    $t1,$t2,endfor  # for(ptarray = array; ptarray < (array + nelem);pta
rray++)
                                # {
        l.d     $f2,0($t1)      #    $f2 = *ptarray
if:     c.le.d  $f2,$f4         #
        bc1f    then
        rem     $t3,$t0, 2      #    $t3 = i % 2
        beq     $t3, $0, endif  #    if ((*ptarray > 0) || ((i % 2) != 0))
then:                           #    {
        mtc1    $t0,$f8         #
        cvt.d.w $f8,$f8         #
        mul.d   $f8,$f8,$f2     #
        cvt.s.d $f8,$f8         #       f8 = (float) (*ptarray) * i
        add.s   $f0,$f0,$f8     #       sum += (*ptarray) * i
                                #    }
endif:  addi    $t0,$t0,1       #    i++
        addu    $t1,$t1,8       #    ptarray++
        j       for             # }
endfor: jr      $ra             #
```