

Soluzione Es 2 (compito 1) - Versione che usa registri \$s*

```

conta_sost:    addi $t0, $0, 1
               bne $a1, $t0, ricorsione    # if (m != 1) goto ricorsione
               lw  $t0, 0($a0)             # else $t0=A[0]
               beq $t0, $a2, ret1          # if A[0]==k goto ret1
               add $v0, $0, $0             # else return 0
               jr  $ra

ret1:          sw  $a3, 0($a0)              # A[0]=h
               addi $v0, $0, 1             # return 1
               jr  $ra

ricorsione:    addi $sp, $sp, -16
               sw  $ra, 0($sp)
               sw  $s0, 4($sp)
               sw  $s1, 8($sp)
               sw  $s2, 12($sp)
               add $s0, $a0, $0            # $s0=A
               add $s1, $a1, $0            # $s1=m
               srl $a1, $a1, 1             # $a1=m/2
               jal conta_cost              # $v0=conta_sost(A,m/2,k,h)
               add $s2, $v0, $0            # $s2=conta_sost(A,m/2,k,h)
               sll $t0, $s1, 1             # $t0=4*m/2
               add $a0, $s0, $t0           # $a0=&(A[m/2]=A+m/2
               srl $a1, $s1, 1             # $a1=m/2
               jal conta_sost              # $v0=conta_sost(A,m/2,k,h)
               add $v0, $s2, $v0           # +conta_sost(A+m/2,m/2,k,h)

               lw  $ra, 0($sp)
               lw  $s0, 4($sp)
               lw  $s1, 8($sp)
               lw  $s2, 12($sp)
               addi $sp, $sp, 16
               jr  $ra

```

Soluzione Es 2 (compito 1) - Versione che non usa registri \$s*

```

conta_sost:    addi $t0, $0, 1
               bne $a1, $t0, ricorsione    # if (m != 1) goto ricorsione
               lw  $t0, 0($a0)             # else $t0=A[0]
               beq $t0, $a2, ret1          # if A[0]==k goto ret1
               add $v0, $0, $0             # else return 0
               jr  $ra

ret1:          sw  $a3, 0($a0)              # A[0]=h
               addi $v0, $0, 1             # return 1
               jr  $ra

ricorsione:    addi $sp, $sp, -16
               sw  $ra, 0($sp)
               sw  $a0, 4($sp)
               sw  $a1, 8($sp)
               srl $a1, $a1, 1             # $a1=m/2
               jal conta_cost              # $v0=conta_sost(A,m/2,k,h)
               sw  $v0, 12($sp)
               lw  $a0, 4($sp)             # $a0=A
               lw  $a1, 8($sp)             # $a1=m
               sll $t0, $a1, 1             # $t0=4*m/2
               add $a0, $a0, $t0           # $a0=&(A[m/2]=A+m/2
               srl $a1, $a1, 1             # $a1=m/2
               jal conta_sost              # $t0=conta_sost(A,m/2,k,h)
               lw  $t0, 12($sp)           # $v0=conta_sost(A,m/2,k,h)
               add $v0, $t0, $v0

```

```

                                +conta_sost(A+m/2,m/2,k,h)
    lw    $ra, 0($sp)
    addi $sp, $sp, 16
    jr    $ra

```

Soluzione Es 3 (compito 1)

```

Es3:      bne $a1, $0, sum      # if m!=0 goto sum
          add $v0, $0, $0      # else return 0
          jr  $ra

sum:      addi $sp, $sp, -20
          sw   $ra, 0($sp)
          sw   $s0, 4($sp)
          sw   $s1, 8($sp)
          sw   $s2, 12($sp)
          sw   $s3, 16($sp)
          add  $s0, $a0, $0      # $s0=A
          add  $s1, $a1, $0      # $s1=m
          add  $s2, $0, $0      # $s2=i=0
          add  $s3, $0, $0      # $s3=somma=0
loop:     beq  $s2, $s1, exit    # if i=m goto exit
          lw   $a0, 0($s0)      # $a0=A[i]
          jal  radiceq          # $v0=sqrt(A[i])
          add  $s3, $s3, $v0    # somma+=sqrt(A[i])
          addi $s2, $s2, 1      # i++
          addi $s0, $s0, 4      # A++
          j    loop

exit:     lw   $ra, 0($sp)
          lw   $s0, 4($sp)
          lw   $s1, 8($sp)
          lw   $s2, 12($sp)
          lw   $s3, 16($sp)
          addi $sp, $sp, 20
          jr   $ra

```