

```

int count(int *, int, int);
int testlimit(int, int);

int main(void)
{
    static int array[20];
    int i = 0;                                // Use $s0

    print_str("Digite 20 valores inteiros:\n");
    do {
        array[i] = read_int();
        i++;
    } while (i < 20);
    i = count(array, 200, 20)
    print_str("Valores menores que 200:\n");
    print_int10(i);
    return 0;
}

```

```

        .data
str1: .asciiz "Digite 20 valores inteiros:\n"
str2: .asciiz "Valores menores que 200:\n"
        .align 2
array: .space 80

        .text
        .globl main
main:
        addiu    $sp, $sp, -8
        sw       $ra, 0($sp)
        sw       $s0, 4($sp)
        li       $s0, 0
        li       $v0, 4
        la       $a0, str1
        syscall

do:
        li       $v0, 5
        syscall
        la       $t0, array
        sll      $t1, $s0, 2
        addu     $t1, $t0, $t1
        sw       $v0, 0($t1)
        addi     $s0, $s0, 1
        blt      $s0, 20, do
        la       $a0, array
        li       $a1, 200
        li       $a2, 20
        jal      count
        move     $s0, $v0
        li       $v0, 4
        la       $a0, str2
        syscall
        li       $v0, 1
        move     $a0, $s0
        syscall
        lw       $ra, 0($sp)
        lw       $s0, 4($sp)
        addiu    $sp, $sp, 8
        li       $v0, 0
        jr       $ra

# int main(void) {
#
#     i = 0;
#
#     print_str (str1);
#     do
#     {
#
#         array[i] = read_int();
#         i++;
#     } while (i < 20);
#
#     i = count(array, 200, 20);
#
#     print_str (str2);
#
#     print_int10(i);
#
#     return 0;
# }

```

```

int count(int *arr, int max, int count)
{
    int nelem = 0;           // Use $s0
    int i = 0;               // Use $s1
    for (; i < count; i++)
    {
        if (testlimit(*arr, max) == 1) nelem++;
        arr++;
    }
    return nelem;
}

```

```

count:      addiu    $sp, $sp, -24           # int count(int *arr, int max, int count) {
            sw      $ra, 0($sp)
            sw      $s0, 4($sp)
            sw      $s1, 8($sp)
            sw      $s2, 12($sp)
            sw      $s3, 16($sp)
            sw      $s4, 20($sp)
            li      $s0, 0                  # nelem = 0;
            li      $s1, 0                  # i = 0;
            move    $s2, $a0                # int *p = arr;
            move    $s3, $a1                # int aux1 = max;
            move    $s4, $a2                # int aux2 = count;

count_for:  bge      $s1, $s4, count_forend # while (i < aux2)
            lw      $a0, 0($s2)             # {
            move    $a1, $s3
            jal     testlimit               # aux2 = testlimit(*p, aux1);
            beqz    $v0, count_next         #
            addi    $s0, $s0, 1             # if (aux2 == 1) nelem ++;

count_next: addiu    $s2, $s2, 4             # p++;
            addi    $s1, $s1, 1             # i++;
            j       count_for              # }

count_forend: move    $v0, $s0              # return nelem
            lw      $ra, 0($sp)
            lw      $s0, 4($sp)
            lw      $s1, 8($sp)
            lw      $s2, 12($sp)
            lw      $s3, 16($sp)
            lw      $s4, 20($sp)
            addiu   $sp, $sp, 24
            jr      $ra                    # }

```

```

int testlimit(int val, int max)
{
    int bellow = 0;           // Use $t9
    if (val < max) bellow = 1;
    val = 0;
    max = 0;
    return bellow;
}

```

```

testlimit:  li      $t9, 0                  # int testlimit(int val, int max){
            bge     $a0, $a1, testlimit_next # int bellow = 0;
            li      $t9, 1                  # if (val < max)
                                            # bellow = 1;

testlimit_next: li      $a0, 0              # val = 0;
                li      $a1, 0              # max = 0;
                move    $v0, $t9            # return bellow;
                jr      $ra                  # }

```