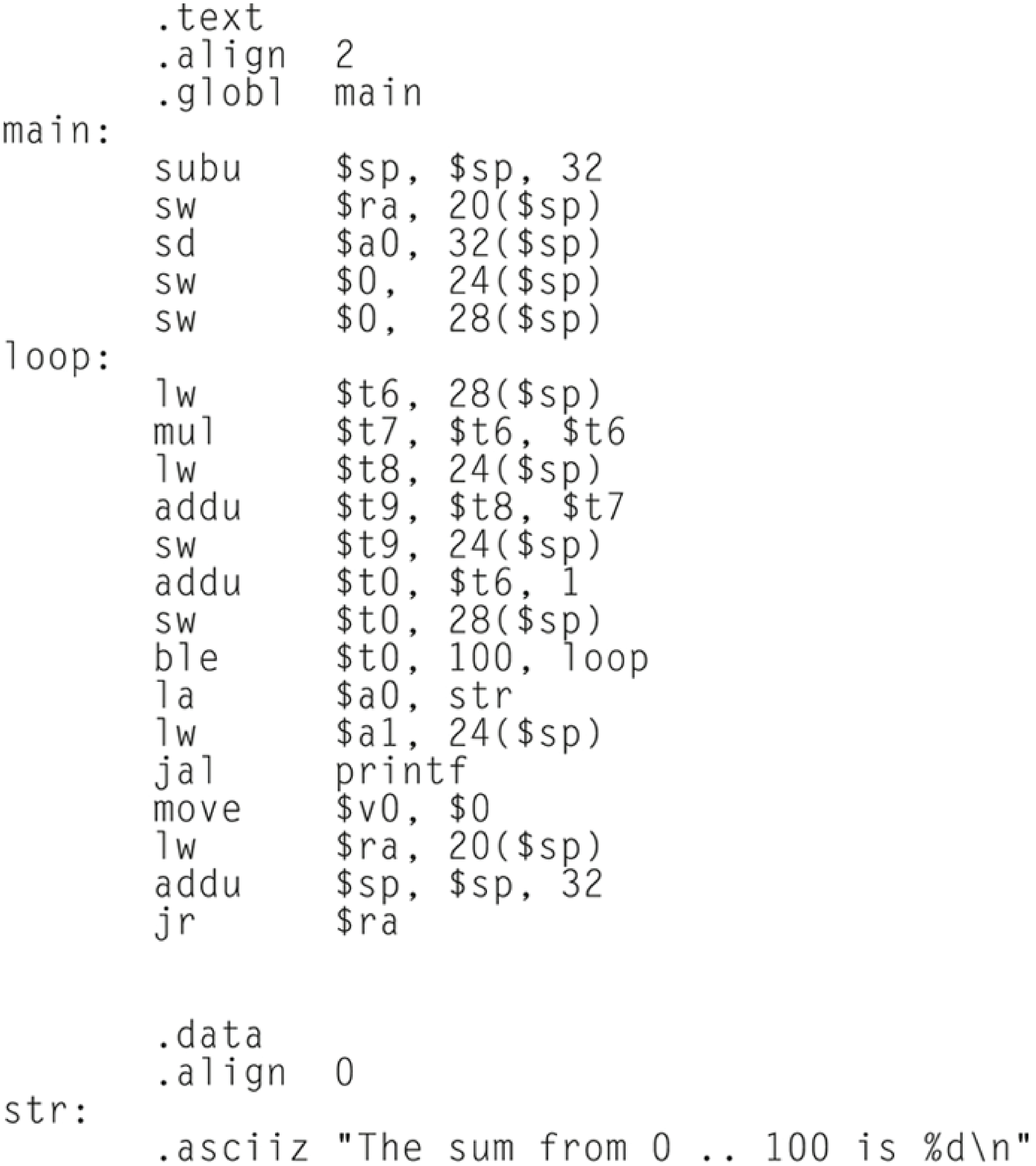

```
        .text
        .align  2
        .globl  main
main:
        subu    $sp, $sp, 32
        sw      $ra, 20($sp)
        sd      $a0, 32($sp)
        sw      $0,  24($sp)
        sw      $0,  28($sp)
loop:
        lw      $t6, 28($sp)
        mul     $t7, $t6, $t6
        lw      $t8, 24($sp)
        addu    $t9, $t8, $t7
        sw      $t9, 24($sp)
        addu    $t0, $t6, 1
        sw      $t0, 28($sp)
        ble     $t0, 100, loop
        la      $a0, str
        lw      $a1, 24($sp)
        jal     printf
        move    $v0, $0
        lw      $ra, 20($sp)
        addu    $sp, $sp, 32
        jr      $ra


        .data
        .align  0
str:
        .asciiz "The sum from 0 .. 100 is %d\n"
```

FIGURE A.1.4 The same routine written in assembly language with labels, but no comments. The commands that start with periods are assembler directives (see pages A-47–49). .text indicates that succeeding lines contain instructions. .data indicates that they contain data. .align n indicates that the items on the succeeding lines should be aligned on a 2n byte boundary. Hence, .align 2 means the next item should be on a word boundary. .globl main declares that main is a global symbol that should be visible to code stored in other files. Finally, .asciiz stores a null-terminated string in memory