

```

public int fibonacci(int n) {
    if(n == 0) return 0;
    else if(n == 1) return 1;
    else return fibonacci(n - 1) + fibonacci(n - 2);
}

```

[REŠITEV: OPTIMIZIRANA]

```

; ---- PODPROGRAM: Iskanje Fibonaccijevih števil ----
.code
.org 0x100
FIB:
    sw 0(r30), r31          ; Na sklad shranimo povratni naslov
    subui r30, r30, #4
    sw 0(r30), r1
    subui r30, r30, #4
    sw 0(r30), r24
    subui r30, r30, #4

    slti r1, r24, #2
    bne r1, END
    subui r24, r24, #1
    call r31, FIB(r0)
    add r1, r28, r0
    subui r24, r24, #1
    call r31, FIB(r0)
    addu r28, r28, r1

RETURN:
    addui r30, r30, #4
    lw r24, 0(r30)
    addui r30, r30, #4
    lw r1, 0(r30)
    addui r30, r30, #4
    lw r31, 0(r30)
    j 0(r31)

END:
    add r28, r0, r24
    j RETURN(r0)

```

```

n:
    ; 0x400-0x7ff
    .data
    .org 0x400
    .word 5

    .code
    .org 0x0

; nastavimo skladovni kazalec na konec podatkovnega segmenta
addui r30, r0, 0x7FC
add r29, r0, r0

main:
    ; funkcija vsota ima le en argument, ki se prenaša preko r24
    lw r24, n(r0)          ; r24 <- n
    ; pokličimo podprogram sestej in povratni naslov shranimo v r31:
    call r31, vsota(r0)
    halt

vsota:
    ; ----- BEGIN VSTOPNA TOCKA: -----
    sw 0(r30), r31          ; shrani svoj povratni naslov na sklad
    subui r30, r30, #4
    sw 0(r30), r29          ; shrani kazalec na okvir klicacega programa
    subui r30, r30, #4
    addu r29, r0, r30       ; R29 <- SP : nastavi kazalec na okvir...sedaj lahko spreminjamo SP
    ; ----- END VSTOPNA TOCKA -----

    ; ----- BEGIN: SHRANI REGISTRE, KI JIH UMAZES, NA SKLAD: -----
    sw 0(r30), r6          ; push r6
    subui r30, r30, #4
    ; ----- END: SHRANI REGISTRE, KI JIH UMAZES, NA SKLAD -----

    ; ----- BEGIN: TELO FUNKCIJE: -----
    add r28, r0, r0        ; predpostavimo, da bomo vrnili ničlo
    beq r24, IZSTOP
    add r6, r24, r0        ; r6 <- trenutni n
    subui r24, r24, #1      ; (n-1)
    call r31, vsota(r0)    ; rekurzivno klici vsota(n-1)
    add r28, r6, r28       ; r28 <- n + vsota(n)
    ; ----- END: TELO FUNKCIJE -----

    ; ----- BEGIN: OBNOVIMO UMAZANE REGISTRE S SKLADU: -----
    addui r30, r30, #4      ; pop r6
    lw r6, 0(r30)
    ; ----- END: OBNOVIMO UMAZANE REGISTRE S SKLADU -----

    ; ----- BEGIN: IZSTOPNA TOCKA: -----
    addu r30, r0, r29      ; pobrisemo lokalne spremenljivke s sklada
    addui r30, r30, #4      ; pop r29
    lw r29, 0(r30)
    addui r30, r30, #4      ; pop r31
    lw r31, 0(r30)
    j 0(r31)              ; vrnitev iz podprograma
    ; ----- END: IZSTOPNA TOCKA -----

```