

```

.title  Fibonnacijeva stevila.
;
; Podprogram izracuna n-to Fibonaccijevo stevilo Fn.
; Ima dva parametra, prvi je n, drugi pa je vrednost
; tesa Fibonaccijevesa stevila.
; Rezultat vrne tudi v registru R0, da ga lahko uporabljamo
; kot podprogram ali funkcijo z imenom IFIB.
; Za Fibonaccijeva stevila velja :
; F0 = 0
; F1 = 1
; Fn = Fn-1 + Fn-2.
;
.entry  ifib    ^m(r2,r3)
movl   4(ap),r3      ; Prvi parameter je n (zaporedno
                    ; stevilo Fibonaccijevesa stevila).
beql   20$         ; n = 0 => vrni rezultat 0.
; Rekurzivni podprogram bomo napisali kot rutino (kontrolno
; prenesemo z bsb ali jsb), da bo izvajanje hitrejse.
jsb    fibonacci
5$:    cmpl      (ap),#1      ; Ali ima procedura IFIB samo en
                    ; parameter (to pomeni, da smo jo
                    ; klicali kot funkcijo)?
beql   10$
movl   r0,08(ap)    ; NE - zapisi rezultat v drugi
                    ; parameter.
10$:   ret          ; DA - rezultat je samo v R0.
20$:   clrl     r0          ; Konec za n = 0.
brb    5$
; Pravi podprogram (routin) za rekurzivno racunanje
; Fibonaccijevish stevil.
; Vhodni parameter je n v registru R3,
; izhodni parameter je vrednost Fibonaccijevesa
; stevila v registru R0.
fibonacci:
decl   r3          ; Ali smo ze prisli do n = 2 (Fn = 1)?
cmpl   r3,#1
bstr   10$        ; NE - poklici podprogram FIBONACCI
                    ; z za 1 manjso vrednostjo v R3.
clrl   r1          ; DA - pripravi vrednosti za F0 in F1
movl   #1,r0       ; v registrih R1 in R0 in pojdi na
brb    20$        ; racunanje F2 (na naslovu 20$).
10$:   jsb     fibonacci
20$:   movl    r0,r2      ; Racunanje Fibonaccijevesa stevila.
addl2  r1,r0         ; V R0 je vedno nazadnje izracunano
movl   r2,r1        ; Fibonaccijevo stevilo, v R1 pa
                    ; predzadnje.
; Z ukazom RSB vracamo kontrolo na ukaz, ki je za klicem
; poprograma FIBONACCI. To ponavljamo tolikokrat, kot smo
; klicali ta podprogram. Sele zanj RSB vrne kontrolo v
; proceuro IFIB.
rsb
.end

```