

1. Data je procedura u programskom jeziku C koja broji koliko ima članova niza cijelih brojeva A koji su veći ili jednaki 10 i manji od 24. Dužina niza A je N.

```
int brojanje(int A[], int N)
{
    int i, br=0;
    for(i=0;i<N;i++)
        if(A[i]>=10 && A[i]<24)
            br++;
    return br;
}
```

### I. Alokacija registara

Pošto se argumenti prosljeđuju preko registara \$4, \$5, \$6 i \$7, a funkcija ima dva ulazna argumenta, njima su pridruženi registri \$4 i \$5, tj.  $A \leftrightarrow \$4$  i  $N \leftrightarrow \$5$ . Promjenljivima **i** i **br** su dodijeljeni registri \$15 i \$16, respektivno.

### II. Kod tijela procedure

```

                add $16, $0, $0      # br=0;
                add $15, $0, $0      # i=0;
Loop:          slt $8, $15, $5       # Ako je i<N onda $8=1
                beq $8, $0, Exit     # Ako je $8=0 izađi iz petlje
                muli $8, $15, 4      # $8=4*i - u $8 prava vrijednost pomjeraja
                add $8, $8, $4       # U $8 adresa i-tog člana niza A[i]
                lw $9, 0($8)         # U $9 član niza A[i]
                slti $10, $9, 10     # Ako je A[i]<10 onda $10=1
                bne $10, $0, Next    # Ako je $10=1 idi na Next
                slti $10, $9, 24     # Ako je A[i]<24 onda $10=1
                beq $10, $0, Next    # Ako je $10=0 idi na Next
                addi $16, $16, 1     # br=br+1
Next:          addi $15, $15, 1     # i=i+1
                j Loop              # go to na Loop
Exit:          add $2, $16, $0      # u $2 vrijednost koju vraća funkcija - return br;
```

### III. Čuvanje sadržaja registara tokom poziva procedura

U proceduri mijenjamo sadržaj registara \$15, \$16, \$8, \$9 i \$10 pa se prije bilo kakve promjene njihov sadržaj mora smjestiti na stek radi čuvanja njihovih originalnih vrijednosti. Registar \$29 sadrži pokazivač na posljednju zauzetu lokaciju steka, pa se on mora umanjiti za  $4 \cdot 5 = 20$  B, kako bi se napravilo prostora za ovih 5 registara:

```
addi $29, $29, -20    # Stek raste od viših adresa ka nižim i otuda -
```

Čuvanje starih vrijednosti u registrima \$15, \$16, \$8, \$9 i \$10:

```
sw $15, 0($29)      # $15 na stek
sw $16, 4($29)      # $16 na stek
sw $8, 8($29)       # $8 na stek
sw $9, 12($29)     # $9 na stek
sw $10, 16($29)    # $10 na stek
```

Obzirom da procedura `brojanje` ne poziva nijednu drugu proceduru, nema potrebe da se \$31 smješta na stek.

Nakon završetka procedure vraćamo sa steka stare vrijednosti registara \$15, \$16, \$8, \$9 i \$10 i usklađujemo pokazivač na stek:

```
lw $15, 0($29)      # sa steka u $15
lw $16, 4($29)      # sa steka u $16
lw $8, 8($29)       # sa steka u $8
lw $9, 12($29)      # sa steka u $9
lw $10, 16($29)     # sa steka u $10
addi $29, $29, 20   # usklađivanje steka
```

Posljednji korak je vraćanje u proceduru koja je pozvala našu proceduru, i to na instrukciju prvu poslije poziva naše procedure:

```
jr $31      # vraćanje u pozivajuću proceduru.
```

### Dakle, ukupan kod bi bio:

```
brojanje: addi $29, $29, -20
          sw $15, 0($29)      # $15 na stek
          sw $16, 4($29)      # $16 na stek
          sw $8, 8($29)       # $8 na stek
          sw $9, 12($29)      # $9 na stek
          sw $10, 16($29)     # $10 na stek
          add $16, $0, $0      # br=0;
          add $15, $0, $0      # i=0;
Loop:     slt $8, $15, $5      # Ako je i<N onda $8=1
          beq $8, $0, Exit     # Ako je $8=0 izađi iz petlje
          muli $8, $15, 4      # $8=4*i - U $8 prava vrednost pomeraja
          add $8, $8, $4       # U $8 adresa i-tog člana niza A[i]
          lw $9, 0($8)        # U $9 član niza A[i]
          slti $10, $9, 10     # Ako je A[i]<10 onda $10=1
          bne $10, $0, Next    # Ako je $10=1 idi na Next
          slti $10, $9, 24     # Ako je A[i]<24 onda $10=1
          beq $10, $0, Next    # Ako je $10=0 idi na Next
          addi $16, $16, 1     # br=br+1
Next:     addi $15, $15, 1     # i=i+1
          j Loop              # go to na Loop
Exit:     add $2, $16, $0      # u $2 vrijednost koju vraća funkcija - return br;
          lw $15, 0($29)      # sa steka u $15
          lw $16, 4($29)      # sa steka u $16
          lw $8, 8($29)       # sa steka u $8
          lw $9, 12($29)      # sa steka u $9
          lw $10, 16($29)     # sa steka u $10
          addi $29, $29, 20   # usklađivanje steka
          jr $31              # vraćanje u pozivajuću proceduru.
```

2. Data je procedura u programskom jeziku C koja za ulazne argumente ima string **S** i dva karaktera, **c1** i **c2**, i koja svaku pojavu tih karaktera u stringu **S** zamjenjuje karakterom '+'.

```
void zamjena(char S[], char c1, char c2)
{
    int i=0;
    while (S[i]!='\0')
    {
        if(S[i]==c1 || S[i]==c2)
            S[i]='+';
        i++;
    }
}
```

**I. Alokacija registara**

$S \leftrightarrow \$4$ ,  $c1 \leftrightarrow \$5$  i  $c2 \leftrightarrow \$6$ . Promjenljivoj  $i$  je dodijeljen registar  $\$15$ .

**II. Kod tijela procedure**

```
      add $15, $0, $0      # i=0;
Loop: add $17, $4, $15     # U $17 adresa A[i]
      lb $18, 0($17)      # U $18 karakter A[i]
      beq $18, $0, Exit   # Ako je $18=0 to je kraj stringa i izadi iz petlje
      beq $18, $5, L1     # Ako je A[i]==c1 idi na L1
      bne $18, $6, Next   # Ako je A[i]!=c2 idi na Next
L1:   addi $18, $0, '+'    # U $18 karakter '+'
      sb $18, 0($17)      # A[i]='+'
Next: addi $15, $15, 1    # i=i+1
      j Loop              # go to na Loop
Exit:
```

**III. Čuvanje sadržaja registara tokom poziva procedure**