

# ZADAĆA 2

## Arhitektura računara

April, 2019.

### Sadržaj

1	Zadatak 1	2
2	Zadatak 2	2
3	Zadatak 3	2
4	Zadatak 4	2
5	Zadatak 5	3
6	Zadatak 6	3
7	Zadatak 7	3
8	Zadatak 8	4
9	Zadatak 9	4

### Zadatak 1

Dat je proizvoljan niz od 10 32-bitnih brojeva. Za svaki element niza primijeniti sljedece pravilo:

- ako je broj paran podijeliti mu vrijednost sa 2, u suprotnom pomnoziti sa 2
- na lokaciju niza “rez” smjestiti apsolutnu vrijednost dobivenog broja

Realizovati bez koristenja funkcija, odnosno koristeći uslove i petlje.

### Zadatak 2

Dat je dio .data sekcije MIPS programa:

```
.section .data
```

```
n: .word 10
```

```
niz1: .byte 3, 4, -44, 34, -34, 0, -43, -3, 34, 55, 34, 34, 33, 4, 54, 4, 5, 0, 0, 34
```

```
niz2: .word 34, 4, 3, 4, 4, 43, 2, 23, 4, 4, 4, 4, -43, 0, 0, 0, 0, 3243, 33, 12
```

```
niz3: .word 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

Napisati MIPS program koji sabira vektore “niz1” i “niz2” te rezultat smjesti u vektor “niz3”.

Na lokaciji “n” je data dimenzija vektora (broj elemenata koje treba uzeti u obzir).

### Zadatak 3

Data je funkcija  $f(x) = \{ 2x + f_1(2x) \text{ za } x \text{ neparno, } 3x + f_2(x, 2x-3) \text{ za } x \text{ parno} \}$ ,  $x$  je cijeli broj.

Izračunati vrijednost funkcije za 10 elemenata niza “brojevi” i rezultate smještati u niz “rezultat”.

Funkcija  $f_1(x) = \{ 5x + 7 \text{ za } x < 10, x - 17 \text{ za } x \geq 10 \}$

Funkcija  $f_2(x, y) = \{ 5x + 2y \text{ za } x \neq 5, y + 4x \text{ za } x = 5 \}$

### Zadatak 4

Za sljedeci asemblerski kod dodati odgovarajuće linije koda da bi program radio ispravno.

Funkcija f\_rezultat je oblika `int f_rezultat (int x)`

```
.section .data
```

```
brojevi: .word 2, 4, 8, 9, 5, 1, 4, 2, 5, 7, 4, 1, 3, 5, 6
```

```
rezultat: .word 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

```
.section .text
```

```
.set noreorder
```

```
.globl main
```

```
main:
```

```
la $t0, niz
```

```
la $t1, rezultat
```

```

addi $t2, $0, 0
j uslov
tijelo:
sll $t3, $t2, 2
add $t4, $t3, $t0
add $t5, $t3, $t1
lw $a0, 0($t4)
jal f_rezultat
sw $v0, 0($t5)
addi $t2, $t2, 1
uslov:
slti $s0, $t2, 13
bne $s0, $0, tijelo
addi $v0, $0, 0
jr $ra
nop

```

### **Zadatak 5**

Funkcija `int test (int x)` je pozvana od strane funkcije `main`, dodati odgovarajuće linije koda da bi program radio ispravno.

```

test:
slti $s3, $a0, 4
beq $s3, $0, else
addi $v0, $0, 0
j kraj
else:
sll $s6, $a0, 1
addi $v0, $s6, 6
kraj:
jr $ra

```

### **Zadatak 6**

Data je funkcija  $f(a, b, c, d) = (a+b) - (c+d) - 6$ ,  $x$  je cijeli broj. Izračunati vrijednost funkcije za  $n$  elemenata niza "brojevi" i rezultate smještati u niz "rezultat".

### **Zadatak 7**

Napisati MIPS program koji za date karaktere (ASCII kodove) u nizu kodovi provjerava da li isti postoje u stringu "tekst" i rezultat upisuje u niz "test".

Provjeru postojanja znaka u stringu realizovati kao proceduru koja bi u C-u imala deklaraciju:

```
int postoji_znak(char *s, char c);
```

Funkcija vraća 0 ako znak c ne postoji u stringu a 1 ako postoji. String zadan pomocu .asciiiz direktive je NULL-terminiran. Program i funkciju provjeriti na sljedećim nizovima:

```
.section .data
```

```
tekst: .ascii "Ovo je neki tekst KOJI SLUži za prOBU. 1 2 3 itd."
```

```
kodovi: .byte 'a', 'O', 'A', '1', 32, 'e', 98
```

```
test: .word 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

## Zadatak 8

Data je rekurzivna funkcija

$$fr(n) = (1+2+3+\dots+n) + fr(n-2)$$
$$fr(0) = 0 \text{ i } fr(1) = 1.$$

Izračunati vrijednost funkcije za prvih 20 prirodnih brojeva i rezultate smještati u niz "rezultat". Za sumu cijelih brojeva  $1+2+3+\dots+n$  koristiti funkciju suma\_n koju također treba realizovati.

## Zadatak 9

Sljedeću rekurzivnu funkciju napisati u assembleru.

```
short test(short a, short b){
    if ( a < 2 ){
        return 1;
    }
    else if( b < 3 ){
        return 1;
    }
    if (a < b){
        return a + test(a-1, b-3);
    }else{
        return a - b + test(a-2, b-1);
    }
}
```