

## ZADACI ZA VJEŽBU:

1. Dat je dio koda u programskom jeziku C:

```
int x1=9;
int x2=1;
while (x1!=0){
    if (x1>=4)
        x2=x1*x2;
    else
        x2=x1+x2;
    x1=x1-2;}
```

Napisati ovaj dio koda u MIPS asemblerskom jeziku pod pretpostavkom da su promjenljivim **x1** i **x2** dodijeljeni registri \$10 i \$11, respektivno.

2. Dat je dio koda u MIPS asemblerskom jeziku:

```
        addi  $16, $0, 5
        addi  $17, $0, 5
        bne   $16, $17, L1
        addi  $15, $16, 2
        j     L2
L1:     sub   $15, $16, $17
L2:     muli  $17, $15, 4
```

Kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$15, \$16 i \$17 nakon njegovog izvršenja.

3. Data je procedura u programskom jeziku C koja broji koliko je elemenata u nizu cijelih brojeva **niz**, ulaznom argumentu procedure, različito od **5** i **-5**. Kada ih detektuje, procedura ih postavlja na 0. Napisati odgovarajući MIPS kod. **(8 poena)**

```
int procedura(int niz[], int N)
{
    int i, brojac=0;
    for (i=0; i<N; i++)
    {
        if(niz[i]!=5 && niz[i]!=-5)
        {
            niz[i]=0;
            brojac++;
        }
    }

    return brojac;
}
```

4. Data je procedura u programskom jeziku C koja treći karakter ulaznog stringa **Rijec** zamjenjuje karakterom **X** ukoliko je taj karakter **N**. Napisati odgovarajući MIPS kod

```
void zamjeni_N(char Rijec[]){
    int j=0;
    while (Rijec[j]!='\0')
        if (j==3) {
            if (Rijec[j]=='N')
                Rijec[j]='X';
            else
                j++;}
    }
```

5. Data je procedura u programskom jeziku C koja za ulazne argumente ima dva cijela broja **X** i **Y** i koja ako je  $X > Y$  vraća njihov proizvod, ako je  $X < Y$  vraća kvadrat broja **X**, a ako je  $X = Y$  vraća njihov zbir. Napisati odgovarajući MIPS kod.

```
int poredjenje(int X, int Y){
    if (X>Y)
        return X*Y;
    elseif (X<Y)
        return X^2;
    else
        return X+Y;
    }
```

6. Dat je dio koda u programskom jeziku C:

```
int a=0;
int b=10;
while (b>=a)
{
    if (b<6)
        b=b-2;
        b--;
    a++;
}
```

Napisati ovaj dio koda u MIPS asemblerskom jeziku pod pretpostavkom da su promjenljivim **a** i **b** dodijeljeni registri \$8 i \$9, respektivno.

7. Dat je dio koda u MIPS asemblerskom jeziku:

```
        addi $8,$0,1
        addi $9,$0,0
loop:   slti $10,$8,4
        beq $10,$0,Exit
        add $9,$9,$8
        addi $8,$8,1
        j loop
Exit:   add $9,$9,$9
```

Što izvršava ovaj dio koda, tj. kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$8 i \$9 nakon njegovog izvršenja.

**8.** Data je procedura u programskom jeziku C koja svaku pojavu karaktera '-' ili '\_' u stringu s zamjenjuje karakterom ' ' i koja kao rezultat vraća broj izvršenih izmjena. String s je ulazni argument procedure. Napisati odgovarajući MIPS kod.

```
int procedura(char s[])
{
    int i=0;
    int n=0;
    while(s[i]!='\0')
    {
        if(s[i]=='_' || s[i]=='-')
        {
            s[i]=' ';
            n++;
        }
        i++;
    }
    return n;
}
```

**9.** Dat je dio koda u programskom jeziku C:

```
int m=1;
int n=5;
while (n<7)
{
    if (n>=3)
        m=m+n;
    n+=1;
}
```

Napisati ovaj dio koda u MIPS asemblerskom jeziku pod pretpostavkom da su promjenljivim **m** i **n** dodijeljeni registri \$17 i \$18, respektivno.

**10.** Dat je dio koda u MIPS asemblerskom jeziku:

```
addi $8,$0,6
addi $9,$0,0
loop: slti $10,$8,4
     bne $10,$0,Exit
     add $9,$9,$8
     addi $8,$8,-1
     j loop
Exit: addi $9,$9,2
```

Što izvršava ovaj dio koda, tj. kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$8 i \$9 nakon njegovog izvršenja.

**11.** Data je procedura u programskom jeziku C koja svaku pojavu velikog slova u stringu s zamjenjuje odgovarajućim malim slovom i koja kao rezultat vraća broj izvršenih izmjena. String s je ulazni argument procedure. Napisati odgovarajući MIPS kod.

```
int procedura(char s[])
{
    int i=0;
    int n=0;
    while(s[i]!='\0')
    {
        if(s[i]>='A' && s[i]<='Z')
        {
            s[i]=s[i]+'a'-'A';
            n++;
        }
        i++;
    }
    return n;
}
```

**12.** Dat je dio koda u programskom jeziku C:

```
int x1=10;
int x2=1;
while (x1!=0){
    if (x1>=5)
        x2=x2*x1;
    else
        x2=x2+x1;
    x1=x1-2;
}
```

Napisati ovaj dio koda u MIPS asemblerskom jeziku pod pretpostavkom da su promjenljivim x1 i x2 dodijeljeni registri \$12 i \$13, respektivno.

**13.** Dat je dio koda u MIPS asemblerskom jeziku:

```
addi $16, $0, 7
addi $17, $0, 7
bne $16, $17, L1
addi $15, $16, 2
j L2
L1: sub $15, $16, $17
L2: muli $17, $15, 4
```

Kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$15, \$16 i \$17 nakon njegovog izvršenja.

14. Data je procedura u programskom jeziku C koja broji koliko je elemenata u nizu cijelih brojeva **niz**, ulaznom argumentu procedure, različito od **1** i **-1**. Kada ih detektuje, procedura ih postavlja na 0. Napisati odgovarajući MIPS kod.

```
int procedura(int niz[], int N)
{
    int i, brojac=0;
    for (i=0; i<N; i++)
    {
        if(niz[i]!=1 && niz[i]!=-1)
        {
            niz[i]=0;
            brojac++;
        }
    }

    return brojac;
}
```

15. Dat je dio koda u MIPS asemblerskom jeziku:

```
addi $19, $0, 5
addi $21, $0, 5
bne $19, $21, Lab1
addi $20, $19, 3
j Lab2
Lab1: sub $20, $19, $21
Lab2: muli $21, $20, 3
```

Kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$19, \$20 i \$21 nakon njegovog izvršenja. (6 poena)

16. Data je procedura u programskom jeziku C koja broji koliko je elemenata u nizu cijelih brojeva **array**, ulaznom argumentu procedure, različito od **2** ili **-2**. Kada ih detektuje, procedura ih postavlja na 0. Napisati odgovarajući MIPS kod. (8 poena)

```
int procedura(int array[], int N)
{
    int i, br=0;
    for (i=0; i<N; i++){
        if(array[i]!=2 || array[i]!=-2)
        {
            array[i]=0;
            br++;
        }
    }
    return br;
}
```

17. a) Objasniti ulogu registara \$2 i \$4.

b) Odrediti sadržaj registra \$10 nakon izvršenja instrukcije slt \$10,\$11,\$12. Sadržaji registara \$11 i \$12 su 4 i 5, respektivno.

**18.** a) Objasniti ulogu registra \$29.

b) Odrediti sadržaj registra \$8 nakon izvršenja instrukcije slti \$8,\$9,10. Sadržaj registra \$9 je 10.

**19.** a) Objasniti ulogu registra \$31.

b) Odrediti sadržaj registra \$8 nakon izvršenja instrukcije slti \$8,\$9,10. Sadržaj registra \$9 je 9.

**20.** a) Objasniti ulogu registara \$3 i \$6.

b) Odrediti sadržaj registra \$10 nakon izvršenja instrukcije slt \$10,\$11,\$12. Sadržaji registara \$11 i \$12 su isti.