

1. Napisati assembly program koji računa zbir prvih n prirodnih brojeva
 - a. Proizvoljan broj n se na početku programa postavlja u neki od dostupnih registara opšte namene, instrukcijom "mov"
 - b. Konačan rezultat treba se naći u nekom od dostupnih registara opšte namene
 - c. Primer: ako je $n = 5$, rezultat je $1+2+3+4+5 = 15$
2. U assembly jeziku implementirati množenje dva broja pomoću sabiranja
 - a. Dva proizvoljna broja koja se množe postavljaju se na početku programa u neke od dostupnih registara opšte namene
 - b. Konačan rezultat treba se naći u nekom od dostupnih registara opšte namene
 - c. Ideja: ako su činioci u registrima `eax` i `ebx`, a suma će se naći u `ecx`, treba `eax` puta vrednost iz `ebx` dodati na `ecx`
 - d. Primer: ako se množe 3 i 5, rezultat se treba dobiti ili kao $3+3+3+3+3$, ili kao $5+5+5$
 - e. Optimizacija: napisati program tako da se petlja vrti minimalan broj puta; ako se množe 3 i 5, petlja treba da ima 3 iteracije bez obzira na to koji se činilac nalazi u kom registru
 - f. Pomoć za optimizaciju: korišćenje "xchgl" instrukcije za zamenu vrednosti dva registra
3. Napisati assembly program koji pronalazi n -ti element Fibonačijevog niza
 - a. Proizvoljan broj n se na početku programa postavlja u neki od dostupnih registara opšte namene, instrukcijom "mov"
 - b. Konačan rezultat treba se naći u nekom od dostupnih registara opšte namene
 - c. Fibonačijev niz – vrednost prvog i drugog elementa niza je 1; svaki naredni element računa se kao zbir prethodna dva; prvih nekoliko elemenata niza:
 $1, 1, 2, 3, 5, 8, 13, 21, 34, 55$
 Na primer, vrednost 7. elementa je $5 + 8 = 13$
 - d. Primer: ako je $n = 6$, rezultat je 8
4. Napisati assembly program koji računa faktorijel broja n
 - a. Proizvoljan broj n se na početku programa postavlja u neki od dostupnih registara opšte namene, instrukcijom "mov"
 - b. Konačan rezultat treba se naći u nekom od dostupnih registara opšte namene
 - c. Primer: za $n = 5$, rezultat je $5! = 5*4*3*2*1 = 120$
 - d. Napomena: za množenje koristiti ranije implementirano množenje pomoću sabiranja
5. U assembly jeziku implementirati deljenje dva broja pomoću oduzimanja, uz čuvanje ostatka
 - a. Dva proizvoljna broja koja se množe postavljaju se na početku programa u neke od dostupnih registara opšte namene
 - b. Konačan rezultat treba se naći u nekom od dostupnih registara opšte namene
 - c. Ideja: ako treba A podeliti sa B, potrebno je A umanjivati za B sve dok B ne postane veće od A; rezultat deljenja jednak je broju izvršenih oduzimanja, a vrednost A po prestanku oduzimanja jednaka je traženom ostatku