

1. Napisati potprogram u asemblerskom jeziku kojim se računa n -ti element Fibonačijevog niza:
 - a. Zaglavlje funkcije (potprograma) dato je u glavni.c fajlu u Z1 folderu
 - b. Potprogram kao povratnu vrednost vraća kod greške – greška nastupa ukoliko dođe do prekoračenja pri sabiranju (za vežbu, možete proširiti zadatak tako da greška nastupa i kada je prosleđeno nevalidno n); potprogram vraća 0 ukoliko nije nastupila greška, a bilo šta drugo ukoliko jeste nastupila
 - c. Potprogram vraća pronađeni n -ti element niza preko parametra *unsigned int* rez* prenetog po adresi
 - d. Fajl *resenje_1.S* sadrži rešenje sa manje uključenih parametara, dok fajl *resenje_2_za_testiranje.S* sadrži rešenje koje se može proslediti skripti za testiranje
2. Napisati potprogram u asemblerskom jeziku kojim se računa n -ti element Fibonačijevog niza korišćenjem **rekurzivnog algoritma**. (Detaljnije o rekurzivnom Fibonačijevom algoritmu možete naći na sledećem linku:

<https://medium.com/launch-school/recursive-fibonacci-method-explained-d82215c5498e>

Osnovna ideja jeste da se n -ti element niza računa po formuli:

$F(n) = F(n-1) + F(n-2)$, za $n > 2$ ukoliko se niz indeksira počev od 1, a $F(1) = F(2) = 1$

- a. Zaglavlje funkcije (potprograma) dato je u glavni.c fajlu u Z2 folderu
 - b. Potprogram rezultate vraća obrnuto od zadatka 1 – pronađeni n -ti element vraća se kao povratna vrednost potprograma, dok se greška vraća preko parametra prenetog po adresi
 - c. **Pomoć** – postoje tri važne celine u ovom potprogramu:
 - i. Postavljanje uslova izlaska iz rekurzije – treba ga postaviti na vrhu, kako bi se rekurzija na vreme prekinula; uslov se ogleda u proveru da li je trenutno $n > 2$, i da li treba ulaziti u dalju rekurziju
 - ii. Poziv potprograma za $F(n-1)$
 - iii. Poziv potprograma za $F(n-2)$ i sabiranje rezultata sa rezultatom poziva $F(n-1)$
 - d. **Napomena:** rešenje dostupno na ACS-u ne sadrži kod koji uzima u obzir grešku, pa ga treba proširiti kako bi odgovarao priloženom .c fajlu
3. Napisati potprogram u assembleru kojim se pokomponentno sabiraju dva niza a i b , a rezultat sabiranja smešta se u niz c ($c[0] = a[0] + b[0]$, $c[1] = a[1] + b[1]$, ..., $c[n] = a[n] + b[n]$):
 - a. Elementi sva tri niza su u dvostrukoj preciznosti
 - b. Zaglavlje funkcije (potprograma) dato je u glavni.c fajlu u Z3 folderu
 - c. Sva tri niza prosleđena su potprogramu kao parametri preneti po adresi
 - d. Potprogram kao povratnu vrednost vraća kod greške; greška nastupa ukoliko dođe do prekoračenja pri sabiranju; potprogram vraća 0 ukoliko nije bilo greške, a bilo šta drugo ukoliko jeste

Napomena: skriptu za testiranje možete pokrenuti tako što joj samo prosledite svoj potprogram, odnosno .S fajl (**ne treba prosleđivati .c fajl**) – *./testiraj.sh moje_resenje.S*. Ukoliko dobijete grešku koja kaže da nemate privilegije za izvršenje skripte, otvorite terminal u folderu gde je skripta za testiranje i unesite komandu:

chmod +x testiraj.sh