

1. Napisati asemblerski potprogram za konverziju iz znakovnog oblika (ceo neoznačeni oktalni broj) u interni oblik:
 - a. Direktorijum *oktalni*
 - b. Potpis funkcije: *unsigned int oct_str_to_inter(char* str, unsigned int* greska)*
 - c. Povratna vrednost – interni oblik broja
 - d. Dopuniti potprogram proverom grešaka:
 - i. Pojava cifre van dozvoljenog opsega u ulaznom stringu, kao i prazan string, vraćaju grešku 1
 - ii. Prekoračenje kapaciteta je greška 2
2. Proširiti prvi zadatak tako da radi sa označenim brojevima:
 - a. Direktorijum *oktalni-označeno*
 - b. Uneti broj, odnosno string, može imati predznak + ili -
 - c. Potpis funkcije: *int oct_str_to_inter(char* str, unsigned int* greska)*
 - d. Greške su iste kao u prvom zadatku
 - e. Pojašnjenje prekoračenja i provere istog:
 - i. Nakon obrade znaka u ulaznom stringu, unos se može obrađivati kao i u prvom zadatku, kao da je uvek u pitanju pozitivan neoznačen broj. Dakle, može se koristiti provera carry flag-a za prekoračenje opsega. Ako program ne bude prekinut zbog ovakvog prekoračenja, već zbog nailaska na karakter \0, treba ispitati vrednost rezultata i predznak u korisničkom unosu. Kako C program očekuje označen broj (povratna vrednost je *int*, a ne *unsigned int*), to znači da će C program rezultat (vraćenu vrednost) posmatrati kao negativan broj ukoliko mu je najviši bit postavljen na 1. Ovo dalje znači da 1 sme da se nađe na najvišem bitu u rezultatu **isključivo ako je korisnik prosledio string sa negativnim predznakom**. Sa druge strane, kao što je već rečeno, unos se obrađuje kao neoznačen (testira se carry flag) i ovo znači da rezultat u toku obrade unosa može da naraste do vrednosti koja ima 1 na najvišem bitu, iako je korisnik uneo string bez predznaka, ili sa predznakom +, odnosno pozitivan broj. Sve ovo znači da po završetku obrade unosa, ukoliko nije nastupila nikakva druga greška, moramo da ispitamo da li je na najvišem bitu dobijenog rezultata 1. Ukoliko jeste, ispitujemo predznak unosa, i ukoliko je negativan vraćamo u C program dobijeni rezultat. Ukoliko je ipak najviši bit 1, a predznak nije negativan, znači da je u pitanju greška.
3. Napisati asemblerski potprogram za konverziju razlomljenog broja predstavljenog nizom znakova u oktalnom brojnom sistemu u interni oblik:
 - a. Direktorijum *oktalni-razlomljeno*
 - b. Potpis funkcije: *unsigned int oct_str_float_to_inter(char* str, unsigned int* greska)*
 - c. Smatrati da se na početku stringa uvek nalazi 0.
 - d. Povratna vrednost je interni oblik broja
 - e. Greška se postavlja na 1 ukoliko se u stringu nalazi nevalidan karakter, ili ukoliko se prosledi prazan string
4. Uvesti parametar *baza* u 2. zadatak i prepraviti da radi sa bilo kojom bazom u opsegu 2-36:
 - a. Direktorijum *str-to-inter-baza*
 - b. Podržati obradu stringa i sa malim i sa velikim slovima
 - c. Potpis funkcije: *int str_to_inter(char* str, unsigned int* greska, unsigned int baza)*
 - d. Preporuka: pogledati kod iz rešenja, ali napisati i sopstvenu implementaciju, ukoliko je dostupno rešenje konfuzno

5. Proširiti zadatak 3 tako da radi sa označenim brojevima (string može imati predznak + ili -) i da prima parametar *baza* koji označava bilo koju bazu u opsegu 2-36 (nema dostupnog rešenja)
6. Napisati potprogram za konverziju iz znakovnog oblika rimskog broja u interni:
 - a. Nema rešenja, ali je priprema u direktorijumu *rimski*
 - b. Za detalje implementacije pogledati uputstvo i C program u praktikumu (poglavlje 10.4)
7. Pogledati direktorijum *sabiranje*; unutra postoji pripremljen sabiranje.S fajl, sa objašnjenjem onoga što treba uraditi u zadatku