

Zadaci sa dostupnim rešenjem

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. Potprogram proverava greške:
 - i. pojava cifre van dozvoljenog opsega u ulaznom stringu, kao i prazan string, vraćaju grešku 1,
 - ii. prekoračenje kapaciteta vraća grešku 2.
2. Proširiti prvi zadatak tako da radi sa označenim brojevima:
 - a. Direktorijum *oktalni-oznaceno*.
 - b. Uneti broj, odnosno string, može imati predznak + ili -, ili može biti bez predznaka.
 - 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. Može se smatrati da u zadatku korisnik unosi predznak (ili ništa), nakon čega unosi apsolutnu vrednost broja. U tom slučaju, korisniku je dozvoljeno da unese nakon predznaka samo vrednost koja je u opsegu pozitivnih brojeva, odnosno vrednost koja ima nulu na krajnjem levom bitu. Jedina dozvoljena vrednost sa jedinicom na krajnjem levom bitu je ona koja ima sve nule na ostalim bitovima, što je apsolutna vrednost najnegativnijeg broja iz opsega (na primer, za 3-bitne podatke, 100 je negativna vrednost, čija je apsolutna vrednost predstavljena kao neoznačeni broj takođe 100 po algoritmu komplementa dvojke, odnosno: 100 ---invertovanje---> 011 ---dodavanje jedinice---> 100). Zbog toga, algoritam radi tako što koristi *jo* za detekciju prekoračenja, jer će se *overflow flag* postaviti onda kada se u rezultatu, odnosno u *eax*, prvi put nađe vrednost sa krajnjim levim bitom postavljenim na 1. Nakon *overflow-a*, treba ispitati da li se radi o jedinom dozvoljenom broju sa jedinicom na krajnjem levom bitu, odnosno o broju sa jedinicom skroz levo i svim ostalim nulama, kao i da li je korisnik uneo minus kao predznak, jer je ova vrednost dozvoljena samo kod negativnih brojeva. Konkretni koraci ovih provera objašnjeni su u komentarima u kodu.
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)*

Zadaci bez dostupnog rešenja

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