

# VEŽBA 5

Danas se za razne računске zadatke rutinski koriste tipovi podataka u pokretnom zarezu. Oni su u jeziku C podržani tipovima `float`, `double` i `long double`. Iako je njihovo korišćenje jednostavno, funkcionalnost koju unose zahteva značajne hardverske i softverske resurse. Drugim rečima, zahtevaju složen hardver (FPU – *engl. Floating Point Unit*) ili složene biblioteke potprograma koje zauzimaju puno prostora u programskoj memoriji i izvršavaju se relativno sporo. U mikrokontrolerskim sistemima FPU obično nije prisutan, a ostali pomenuti resursi su skromni i programer ne može sebi da priušti rad sa brojevima u pokretnom zarezu.

Da bi se podržale matematičke operacije koje zahtevaju rad sa brojevima koji nisu celi, a da se pri tom ne koriste brojevi u pokretnom zarezu, moguće je koristiti brojeve sa fiksnim zarezom. Njihova velika prednost je što se mogu efikasno izvoditi na hardveru koji je predviđen za rad sa celim brojevima. Istovremeno, mana im je ograničen opseg brojeva što zahteva od programera određenu pažnju pri osmišljavanju algoritma i izvođenju operacija.

## Brojevi u fiksnom (nepokretnom) zarezu

Pretpostavimo da za zapisivanje decimalnog broja (broja zapisan u sistemu osnove 10) imamo na raspolaganju  $p$  celobrojnih i  $q$  razlomljenih mesta:

`aaaa.bb`

U datom primeru,  $p$  je 4, a  $q$  2. Konkretno vrednosti,  $p$  i  $q$  treba izabrati u skladu sa problemom koji se rešava, čime će biti određen i najmanji i najveći broj koji se može predstaviti ( $\pm 9999.99$ ), kao i preciznost (najmanja razlika između dve vrednosti – u primeru  $0.01$ ).

### Predstavljanje brojeva

Sve brojeve u fiksnom zarezu predstavljamo kao cele brojeve

`aaaabb`

što je ceo broj, a dobija se množenjem broja u fiksnom zarezu sa  $10^q$ . Obrnuto, broj u fiksnom zarezu dobija se deljenjem sa  $10^q$ , odnosno množenjem sa  $10^{-q}$ .

### Sabiranje i oduzimanje brojeva

Pri sabiranju i oduzimanju, broj decimalnih mesta se ne menja

`aaaa.bb + cccc.dd = eeeee.ff`

Iz toga proizilazi da se sabiranje obavlja prostim sabiranjem celih brojeva koji predstavljaju brojeve u fiksnom zarezu. Može doći do pojave dodatne cifre sa leve strane koja nema gde da se zabeleži zbog čega je moguće da dođe do gubitka tačnosti rezultata (prekoračenja):

`aaaabb*10^(-q) + ccccd*10^(-q) = eeeeeff*10^(-q)`

### Množenje brojeva

Posmatrajmo prvo celobrojnu predstavu brojeva:

$$aaaabb \cdot 10^{(-q)} * cccdd \cdot 10^{(-q)} = eeeeeefffff \cdot 10^{(-2q)}$$

Da bi se rezultat pravilno interpretirao, neophodno ga je prvo podeliti sa  $10^q$  :

$$(aaaabb \cdot 10^{(-q)} * cccdd \cdot 10^{(-q)}) \cdot 10^{(-q)} = eeeeeeff \cdot 10^{(-q)}$$

U tom postupku dve krajnje desne cifre će biti izgubljene. Da bi se gubitak sveo na najmanju moguću meru treba obaviti zaokruživanje što će se postići dodavanjem  $5 \times 10^{q-1}$  za pozitivne ili  $-5 \times 10^{q-1}$  za negativne brojeve (polarizacija radi zaokruživanja, *engl. bias*).

### Deljenje brojeva

$$(aaaabb \cdot 10^{(-q)}) / (cccd \cdot 10^{(-q)}) = eeee$$

U rezultatu nedostaje razlomljeni deo, trebalo bi da bude oblika  $eeeeff \cdot 10^{(-q)}$ . Da bi se to postiglo, može se uraditi sledeće:

$$aaaabb \cdot 10^q / cccdd = (eeeeff \cdot 10^{(-q)}) \cdot 10^q$$

Da je računanje bez gubitaka,  $10^{(-q)}$  bi se skratilo. Ovako,  $10^{(-q)}$  uz deljenika obezbeđuje da je on uvek veći od delitelja, pa neće doći do gubitka značajnih cifara.

## Tipovi garantovane dužine

Programski jezik C obezbeđuje heder fajl `stdint.h`. Ovo omogućuje upotrebu celobrojnih tipova kod kojih je dužina reči u bitima zagantovana. Može se birati između označene i neoznačene varijante.

označeni celi brojevi	broj bita
<code>int8_t</code>	8
<code>int16_t</code>	16
<code>int32_t</code>	32
<code>int64_t</code>	64
neoznačeni celi brojevi	broj bita
<code>uint8_t</code>	8
<code>uint16_t</code>	16
<code>uint32_t</code>	32
<code>uint64_t</code>	64

## Zadaci

Za sve proračune koristiti format sa fiksnim zarezo sa 4 decimalna mesta, tj. treba da je  $q=4$ . Za skladištenje celog broja u kom se broj u fiksnom zarezu predstavlja koristiti celobrojni označeni tip sa 32 bita.

1. Napisati funkciju za ispis broja u fiksnom zarezu:

```
void printfix(int32_t fixedpoint, unsigned decplc);
```

Drugi parametar predstavlja broj decimalnih mesta, tj. konstantu  $q$ . Funkcija ispisuje broj na tekućem mestu ispisa bez prelaska u novi red. Testirati pozivanjem iz glavnog programa. Za konstruisanje brojeva u fiksnom zarezu mogu se koristiti makro funkcije sa kraja ove pripreme

za vežbe.

2. Napisati program koji od korisnika traži unos broja  $x$ . Program treba da izračuna i ispiše vrednost  $1/x$ . Rezultat treba da ima 4 decimalna mesta. Koristiti funkciju za ispis. Proveriti rezultat korišćenjem kalkulatora.
3. Napisati program koji od korisnika traži unos broja  $x$ . Program treba da izračuna i ispiše vrednost  $0,2*(x+1,5)$ . Rezultat treba da ima 4 decimalna mesta. Koristiti funkciju za ispis. Proveriti rezultat korišćenjem kalkulatora. Koliko je rezultat tačan i zašto postoje razlike?

Dodatni neobavezni zadatak:

4. Koristeći Vavilonsku metodu računanja kvadratnog korena, napisati program za računanje kvadratnog korena zadatog celog broja koristeći isključivo brojeve u fiksnom zarezu.

## Pomoć za rešavanje zadataka

### Vavilonska metoda računanja kvadratnog korena

Metoda je iterativna,  $x_{n+1}$  se računa na osnovu  $x_n$ , a rezultat je postignut kada se razlika između dva uzastopna rezultata smanji na nivo manji od preciznosti ili unapred zadate tolerancije. Početna vrednost je  $x_0$  i može se uzeti da iznosi polovinu broja  $s$  čiji se koren traži:

$$x_{n+1} = \frac{1}{2} \left( x_n + \frac{s}{x_n} \right), \quad x_0 = \frac{s}{2}$$

### Makro funkcije za konstrukciju broja u fiksnom zarezu

```
#define MKFIX(i, f, pq) ((i)*pq + f)
```

$pq$  treba da ima vrednost  $10^q$ , gde je  $q$  broj predviđenih decimalnih mesta ( $q$  u uvodu vežbe).  $i$  je celobrojni deo, a  $f$  razlomljeni deo broja.

Npr.

```
uint32_t fixnum;
...
fixnum = MKFIX(3, 1416, 10000);
// fixnum sadrzi celobrojnu vrednost 31416
// u pokretnom zarezu sa 4 decimalna mesta to je 3.1416
```

### Ispis broja u fiksnom zarezu

Vrednost  $pd$  se računa kao na osnovu broja decimalnih mesta kao  $10^q$  u odgovarajućoj petlji ako je broj decimalnih mesta  $q$  zadat.

Celobrojni deo jednak je rezultatu deljenja sa  $pd$ .

Razlomljeni deo jednak je ostatku deljenja sa  $pd$ .

Za ispis se može koristiti funkcija `printf` sa formatom `"%d.%04d"`. Bitno je da se decimalni deo ispiše sa vodećim nulama. Pošto je broj decimalnih mesta programabilan, mora se ostaviti mogućnost da se broj 4 promeni. Sledeći primer daje predlog kako to da se izvede.

```
char fmt[] = "%d.%04d"
```

```
fmt[5] = '0'+br_dec_mesta;  
printf(fmt, celi, decimalni);
```

Treba napomenuti da predloženi fragment radi dobro za broj decimalnih mesta od 1 do 9.