Invarijanta petlje – dokaz ispravnosti algoritma

1. Konstruisati algoritam koji konvertuje prirodan broj u njegov oktalni zapis. Dokazati korektnost tog algoritma.

Algoritam Okt cifre(n);
Ulaz: n (prirodan broj)
Izlaz: b (niz oktalnih cifara broja n)

```
t  = n;
k  = 0;
while (t > 0) {
  k = k + 1;
  b[k] = t % 8; t = t / 8;}
```

Dokaz korektnosti algoritma:
invarijanta petlje, tj. induktivna hipoteza: Neka je m broj cije su oktalne cifre dobijene u k prvih prolazaka kroz petlju, pri cemu je $b[1]$ najnizi bit.
Tada je $t\cdot 8^k + m = n$
Baza indukcije: Za $k = 0$ je $t = n$ i $m = 0$, te je ovo tvrđenje tačno.
Neka je induktivna hipoteza tačna za neko
$k != 0$. U narednom prolasku kroz petlju dobijaju se nove vrednosti $b[k+1] = t \% 8$, $t' = t / 8$, i
$m = b[k+1]\cdot 8^k + m = (t \% 8)\cdot 8^k + m$, te je
$t'\cdot 8^{k+1} + m' = (t / 8)\cdot 8^{k+1} + (t \% 8)\cdot 8^k + m = 8^k (8 \cdot (t / 8) + (t \% 8)) + m = 8^k \cdot t + m = n$
zbog induktivne hipoteze.

Dakle, dokazano je da izraz $t\cdot 8^k + m$ ne menja vrednost od prolaska do prolaska kroz petlju, odnosno da je on invarijanta petlje.

Da li algoritam terminira rad, tj. da li će u nekom trenutku biti $t = 0$?
Niz $t_k = t / 8$ je mnotono opadajući niz prirodnih brojeva, te je po algebarskom principu minimalnog elementa ograničen odozdo nulom.
Kako je posle poslednjeg prolaska kroz petlju $t = 0$, u tom trenutku će biti $m = n$, čime je korektnost algoritma dokazana.

2. Konstruisati algoritam koji konvertuje prirodan broj u njegov binarni zapis. Dokazati korektnost tog algoritma.
REŠENJE: Videti poglavlje 1.10 u knjizi *Algoritmi,* Miodrag Živković

3. Konstruisati algoritam koji konvertuje binarni broj u njegov dekadni zapis. Dokazati korektnost tog algoritma pomocu invarijante petlje.

REŠENJE: Pogledati rešenje zadatka 1.23 u knjizi *Algoritmi,* Miodrag Živković

4. Dokažite korektnost sledećeg algoritma za nalaženje NZD (x,y) tj. najvećeg zajedničkog delioca prirodnih brojeva x, y

Algoritam NZD(x, y)
ulaz: x, y

```
izlaz: d /* NZD(x,y) */
{
d =x;
if (y <d) d=y; /* d = min(x, y); */
while ( (x %d !=0) || (y %d != 0) ) d = d − 1;
return d;
}
```

RESENJE:

Definicija d = NZD(x, y) ako je d delilac za x i y, i svaki drugi delilac za x i y je manji od d.
Primer: NZD(24*,* 16) = 8
Napomena: $1 \le NZD(x, y) \le \min\{x, y\}$
Tvrđenje: **while** petlja se završava (nije beskonačna).
Dokaz: Uslov izlaska iz ciklusa proverava da li d ne deli x ili d ne deli y. Dok god je to tačno, ponavlja se telo ciklusa.
U suprotnom izlazi se iz ciklusa.
Uočimo da pre ulaska u ciklus važi d = min(x, y). U svakoj iteraciji ciklusa d se umanjuje za 1.
U najgorem slučaju d će eventualno postati 1, i onda uslov izlaska iz ciklusa neće biti true, jer sigurno važi da
(x %1 ==0) && (y %1==0). Zbog toga ciklus se zavrsava.
Tvrđenje: Invarijanta petlje: $d \ge NZD(x, y)$.
Dokaz: Ovo svojstvo važi pre prve iteracije ciklusa, jer inicijalno d = min(x, y), a znamo iz algebre da važi $\min\{x, y\} \ge$ NZD(x, y).
Dalje, potrebno je pokazati da ako svojstvo $d \ge NZD(x, y)$ važi pre iteracije, ono važi i nakon iteracije.
Pretpostavimo da pre iteracije $d \ge NZD(x, y)$ i neka je izvršena iteracija ciklusa, tj. testirani uslov izlaska iz ciklusa je tačan i izvršena je naredba d = d − 1;
Kako je uslov izlaska iz ciklusa tačan, onda stara vrednost za promenljivu d nije delilac i od x i y, tako da stara vrednost za promenljivu d nije jednaka NZD(x, y).
Kako nova vrednost za d je manja od stare vrednosti za 1, sledi da nova vrednost za d je opet veća ili jednaka od
NZD(x,y) pre sledeće iteracije. Time je završen argument da $d \ge NZD(x, y)$ je invarijantno svojstvo while ciklusa
u NZD algoritmu.

Na kraju, istražimo šta se događa nakon izlaska iz while ciklusa. Iz ciklusa se izlazi kada d jeste delilac i od x i od y.
Dakle, važi da $d \le$ NZD(x, y) (*) kada se izađe iz ciklusa. No, kako važi invarijanta petlje $d \ge$ NZD(x, y) (**), onda
zbog (*) i (**) važi da d = NZD(x, y).
I ova vrednost vraća kao rezultat rada algoritma NZD, što znači da algoritam je korektan.