

## Programiranje 2 — tretji par domačih nalog

- ① Naj bo  $I = \mathbb{Z}^+ \cap [a, b]$  interval v množici pozitivnih celih števil,  $f: I \rightarrow \mathbb{Z}^+$  pa strogo monotonno naraščajoča funkcija (za vse  $x \in I$  in  $y \in I$  velja  $x < y \implies f(x) < f(y)$ ). Funkcija  $f(x) = x^2$ , denimo, je monotonno naraščajoča na poljubnem intervalu  $I$ .

Inverz funkcije  $f$  pri vrednosti  $y$  (oznaka:  $f^{-1}(y)$ ) je taka vrednost  $x^* \in I$ , da velja  $y = f(x^*)$ . Na primer, če je  $f(x) = x^2$ , potem je  $f^{-1}(25) = 5$ .

Napišite funkcijo

```
long inverz(long y, long a, long b)
```

ki za funkcijo  $f$ , definirano na intervalu  $[a, b]$ , vrne vrednost  $f^{-1}(y)$ . V vseh testnih primerih velja, da inverz na podanem intervalu obstaja, velja pa tudi  $1 \leq a \leq b \leq 10^{18}$  in  $1 \leq f(a) \leq f(b) \leq 10^{18}$ . V 50% testnih primerov velja  $1 \leq a \leq b \leq 10^6$ .

Lahko predpostavite, da je funkcija  $f$  definirana s sledečo C-jevsko funkcijo:

```
long f(long x)
```

Dopolnite in oddajte datoteko `inverz.c`. Datoteka naj vsebuje funkcijo `inverz`, funkcij `f` in `main` pa *ne sme* vsebovati. Funkciji `f` in `main` bosta definirani v posameznih testnih datotekah (`test*.c`).

Izhodiščno različico datoteke `inverz.c`, primer testne datoteke in deklaracijsko datoteko `inverz.h` (ta služi kot vez med testno datoteko in datoteko `inverz.c`) najdete v paketu *Izhodiščne datoteke* na spletni učilnici. V datoteki `inverz.c` so zapisana tudi navodila za prevajanje in poganjanje.

- ② Naj bo  $I = \mathbb{Z}^+ \cap [a, b]$  interval v množici pozitivnih celih števil, funkcija  $f: I \times I \rightarrow \{0, 1\}$  pa naj predstavlja dvojiško relacijo. Števili  $x \in I$  in  $y \in I$  sta v relaciji natanko tedaj, ko je  $f(x, y) = 1$ . Relacija  $f$  je na intervalu  $I$

- *tranzitivna*, če za vsak  $x \in I$ ,  $y \in I$  in  $z \in I$  velja  $f(x, y) = 1 \wedge f(y, z) = 1 \implies f(x, z) = 1$ ;
- *antitransitivna*, če za vsak  $x \in I$ ,  $y \in I$  in  $z \in I$  velja  $f(x, y) = 1 \wedge f(y, z) = 1 \implies f(x, z) = 0$ ;
- *netranzitivna*, če obstajajo takšni  $x \in I$ ,  $y \in I$  in  $z \in I$ , da velja  $f(x, y) = 1 \wedge f(y, z) = 1 \wedge f(x, z) = 0$ .

Na primer, na poljubnem intervalu  $I$  je relacija  $(x < y)$  tranzitivna, relacija  $(y = x + 1)$  antitransitivna, relacija  $(x \bmod y = 1)$  pa netranzitivna, ne pa nujno tudi antitransitivna (na primer, na intervalu  $[2, 6]$  je netranzitivna in antitransitivna, na intervalu  $[2, 7]$  pa samo netranzitivna). Relacija  $(y = x^2)$  je na intervalu  $[2, 100]$  antitransitivna, na intervalu  $[1, 100]$  pa zgolj netranzitivna.

Napišite funkcijo

```
int tranzitivnost(int a, int b)
```

ki vrne

- vrednost 1, če je relacija  $f$  na celoštevilskem intervalu  $[a, b]$  tranzitivna in antitransitivna hkrati (da, to je povsem mogoče!);

- vrednost 2, če je relacija  $f$  na celoštevilskem intervalu  $[a, b]$  tranzitivna, pogoji za vrednost 1 pa niso izpolnjeni;
- vrednost 3, če je relacija  $f$  na celoštevilskem intervalu  $[a, b]$  antitransitivna, pogoji za vrednosti 1 in 2 pa niso izpolnjeni;
- vrednost 4, če je relacija  $f$  na celoštevilskem intervalu  $[a, b]$  netransitivna, pogoji za vrednosti 1, 2 in 3 pa niso izpolnjeni.

Lahko predpostavite, da je relacija  $f$  definirana s sledečo C-jevsko funkcijo:

```
bool f(int x, int y)
```

Funkcija vrne `true`, če sta  $x$  in  $y$  v relaciji, in `false`, če nista.

Dopolnite in oddajte datoteko `tranzitivnost.c`. Datoteka naj vsebuje funkcijo `tranzitivnost`, funkcij `f` in `main` pa *ne sme* vsebovati. Funkciji `f` in `main` bosta definirani v posameznih testnih datotekah (`test*.c`).

Izhodiščno različico datoteke `tranzitivnost.c`, primer testne datoteke in deklaracijsko datoteko `tranzitivnost.h` (ta služi kot vez med testno datoteko in datoteko `tranzitivnost.c`) najdete v paketu *Izhodiščne datoteke* na spletni učilnici. V datoteki `tranzitivnost.c` so zapisana tudi navodila za prevajanje in poganjanje.