

## Programiranje 2 — deveti par domačih nalog

- ① V datoteki `naloga1.h` sta podani sledeči deklaraciji:

```
typedef struct Vozlisce Vozlisce;

struct Vozlisce {
    int podatek;
    Vozlisce* naslednje;
};
```

V datoteki `naloga1.c` napišite funkcijo

```
Vozlisce* zdesetakaj(Vozlisce* zacetek, int k),
```

ki v povezanem seznamu, na čigar prvo vozlišče kaže kazalec `zacetek`, ohrani vsako  $k$ -to vozlišče, ostala pa izloči in dealocira. Funkcija naj vrne kazalec na novi začetek seznama oziroma `NULL`, če je nastali seznam prazen. V vseh testnih primerih velja  $k \geq 1$ , vhodni seznam pa ne vsebuje več kot 1000 vozlišč.

Dopolnite in oddajte datoteko `naloga1.c`. V paketu *Izhodiščne datoteke* na spletni učilnici najdete izhodiščno različico te datoteke, datoteko `naloga1.h` in en par testnih datotek.

- ② V datoteki `naloga2.h` so podane sledeče deklaracije:

```
typedef struct A A;
typedef struct B B;
typedef struct C C;

struct A {
    int p;
    B* b;
};

struct B {
    char* q; // nikoli ni NULL
    C* c;
};

struct C {
    bool r;
    A* a;
    B* b;
};
```

V datoteki `naloga2.c` napišite funkcijo

```
int izpisiA(A* a, char* cilj),
```

ki v že alocirano (in dovolj veliko) tabelo, na začetek katere kaže kazalec `cilj`, izpiše bodisi niz `NULL` (če je kazalec `a` enak `NULL`) bodisi niz oblike `{p, b}` (v nasprotnem primeru), pri čemer je `p` izpis vrednosti komponente `p` strukture, na katero kaže kazalec `a`, `b` pa je bodisi niz `NULL` (če je kazalec `b` v tej strukturi enak `NULL`) bodisi niz oblike `{q, c}` (v nasprotnem primeru), pri čemer je `q` niz, na začetek katerega kaže kazalec `q` znotraj strukture, na katero kaže kazalec `b`, `c` pa je bodisi niz `NULL` (če je kazalec `c` v tej strukturi enak `NULL`) bodisi niz oblike `{r, a, b}` (v nasprotnem primeru), pri čemer je `r` vrednost (zapisana kot niz `true` oziroma `false`) komponente `r` strukture, na katero kaže kazalec `c`, `a` in `b` pa sta niza, ki se na že opisan način zgradi na podlagi kazalcev `a` in `b` v tej strukturi.

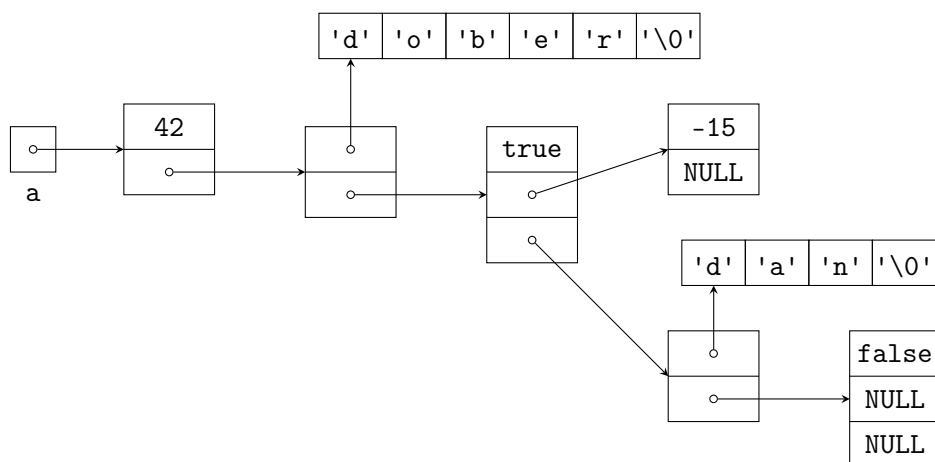
Funkcija naj vrne dolžino niza, izpisanega v tabelo na pomnilniškem naslovu `cilj` (zaključnega znaka `'\0'` seveda ne štejemo).

V vseh testnih primerih je graf medsebojne povezanosti struktur acikličen, dolžina ciljnega niza pa ne presega  $10^4$ .

Na primer, za shemo na sliki 1 bi funkcija v tabelo na pomnilniškem naslovu `cilj` izpisala niz

```
{42, {dober, {true, {-15, NULL}, {dan, {false, NULL, NULL}}}}
```

vrnila pa bi vrednost 62.



Slika 1: Primer, obravnavan v besedilu.

Dopolnite in oddajte datoteko `naloga2.c`. V paketu *Izhodiščne datoteke* na spletni učilnici najdete izhodiščno različico te datoteke, datoteko `naloga2.h` in en par testnih datotek.

Namig: pomagajte si s funkcijo `sprintf`. Morda vam bodo koristile tudi vrednosti, ki jih vrača.