

2. (Opomba)

Pri pisanju besedila uporabimo oklepaje, če želimo zapisati kakšno opombo (ali kaj drugega manj pomembnega). Včasih tudi znotraj opombe povemo kaj še manj pomembnega (torej manj pomembnega od opombe (ki že sama ni pomembna), a še vedno zanimivega); v takem primeru lahko uporabimo gnezdene oklepaje.

Gnezdenje oklepajev lahko načeloma neomejeno stopnjujemo — možno je imeti oklepaje v oklepajih v oklepajih v oklepajih . . . v oklepajih. Če se nam pri branju takšnega teksta mudi, lahko preskočimo vse dele besedila, ki so gnezdeni vsaj k korakov globoko, pri čemer je vrednost k odvisna od tega, kako hudo se nam mudi.

Napiši podprogram `SamoPomembno(s, k)`, ki sprejme niz s s tovrstnim besedilom in naravno število k , nato pa na standardni izhod izpiše to, kar ostane od besedila s , če iz njega pobrišemo vso vsebino, zapisano v k ali več oklepajih. Iz besedila naj bodo pri tem pobrisani tudi prazni oklepajski pari, t.j. takšni, ki ne vsebujejo med oklepajem in zaklepajem nobenega drugega znaka, niti presledka. Pozor, oklepajski par je lahko prazen tudi zato, ker smo iz njegove notranjosti brisali druge prazne ali pregloboko gnezdene oklepajske pare.

Predpostaviš lahko, da so oklepaji v besedilu smiselno postavljeni; tako na primer niz „foo)bar(baz)qux“ ni veljaven vhodni podatek.

Primer: ob klicu

```
SamoPomembno("ja (ja(ja(a b)(tri)) r((c d)(ef ghi))tk() ena) nič", 2);
```

se mora izpisati:

```
ja (ja(ja) rtk ena) nič
```

Tvoj podprogram naj bo takšne oblike:

```
void SamoPomembno(char *s, int k);           /* v C/C++ */
void SamoPomembno(string s, int k);         // v C++
public static void samoPomembno(String s, int k); // v javi
procedure SamoPomembno(s: string; k: integer); { v pascalu }
def SamoPomembno(s, k): ...                 # v pythonu
```

Če ti je ta naloga pretežka, lahko rešiš naslednjo lažjo različico: pobriši vso vsebino, zapisano v k ali več oklepajih, ne briši pa praznih oklepajskih parov. Rešitev te lažje različice dobi 10 točk (namesto 20).