

Rešitev oddajte prek Učilnice. Za rešitev naloge lahko dobite določeno število točk, **tudi če ne prestane testov**. Funkcija, ki prestane vse teste, **še ni nujno pravilna**. Upošteva se tudi kvaliteta rešitve.

Dovoljena je uporaba vseh materialov na Učilnici in druge literature na poljubnih medijih. Prepovedan je dostop do vseh drugih spletnih strani in vsaka oblika komunikacije, razen s profesorjem oz. asistentom.

## 1. Stiskanje

Napiši funkcijo `stisni(s)`, ki prejme niz, sestavljen iz zaporedja znakov A, C, G in T ter vrne niz, v katerem so zaporedne ponovitve istega znaka nadomeščene s številom ponovitev in znakom, vmes pa so presledki. Tako naj klic `stisni("AAAGTTTTGG")` vrne `"3A 1G 4T 2G"`.

Nato napiši funkcijo `razsiri(s)`, ki počne nasprotno: klic `razsiri("3A 1G 4T 2G")` vrne `"AAAGTTTTGG"`.

## 2. LZW

Napiši funkcijo `lzw_stisni(s, kode)`, ki prejme niz in slovar kod ter vsak znak zamenja s pripadajočo kodo. Klic `lzw_stisni("DAVID", {"E": "00", "A": "01", "I": "10", "D": "110", "V": "111"})` vrne `"1100111110110"` (ker D zamenja z 110, A z 01, V z 111, I z 10 in D z 110). V nizu ni nobenih presledkov; znaki v primeru so podčrtani in razmaknjeni le za lažje branje.

Poleg tega napiši funkcijo `lzw_razsiri(s, kode)`, ki naredi obratno: `lzw_razsiri("1100111110110", {"D": "110", "A": "01", "E": "00", "V": "111", "I": "10"})` vrne `"DAVID"`. Pri tem smeš (moraš!) predpostaviti, da koda nobenega znaka ni enaka začetku koda kakega drugega.

Niz, ki ga moraš razširiti, beri po znakih. Najprej vzameš 1; to ni koda nobenega znaka. Nato dodaš drugo 1. Tudi 11 ni koda nobenega znaka. Dodaš 0. 110 je koda D, torej imaš D. Nato imaš 0. 0 ni koda nobenega znaka. Dodaš 1. 01 je koda A. Če ne razumeš, poskusi zaporedje 1100111110110 najprej prebrati ročno, pa boš videl, da je reč očitna.

## 3. Odkodiraj znak

Kode znakov iz primera v prejšnji nalogi bi lahko predstavili tudi z gnezdenim seznamom. Za gornji primer bi kode shranili kot `[["E", "A"], ["I"], ["D", "V"]]`. Pri takem seznamu namreč velja, recimo, da je `kode[1][1][0]` enako D – koda "D"-ja pa je ravno "110", kakor gredo gornji indeksi, `[1][1][0]`.

Napiši nerekurzivno funkcijo `odkodiraj(koda, kode)`, ki prejme kodo znaka in vrne pripadajoči znak; `odkodiraj("110", [["E", "A"], ["I"], ["D", "V"]])` vrne "D".

## 4. Odkodiraj znak (R)

Napiši funkcijo `odkodiraj_rek(koda)`, ki naredi isto kot prejšnja, le da je rekurzivna.

## 5. Kodirnik

Napiši razred `Kodirnik`, ki zakodira besedilo tako, da vsak znak zamenja z zaporedno številko, ki jo dodeljuje v takšnem vrstnem redu, kot se znaki pojavljajo. Besedo `PREPOVED` shrani kot `[0, 1, 2, 0, 3, 4, 2, 5]`, pri čemer velja `P=0, R=1, E=2, O=3, V=4` in `D=5`. Razred ne kodira celega niza naenkrat, temveč mu podajamo vsak znak posebej.

Razred naj ima metode

- `dodaj(self, znak)` doda zakodiran znak. Pri tem uporabi obstoječo kodo, če je znak že nastopal, ali pa mu dodeli novo, če ga vidi prvič. Poglej, kako testi, čisto na koncu, kodirajo "PEPA";
- `kodirano(self)` vrne seznam kod, na primer `[0, 1, 2, 0, 3, 4, 2, 5]`, če kodiramo `PREPOVED`;
- `kode(self)` vrne slovar s kodami, na primer `{"P": 0, "R": 1, "E": 2, "O": 3, "V": 4, "D": 5}`;
- `znaki(self)` vrne slovar z znaki, na primer `{0: "P", 1: "R", 2: "E", 3: "O", 4: "V", 5: "D"}`.

Dodaš lahko poljubne attribute in druge metode, vendar programiraj pametno in ne dodajaj, kar ni potrebno.