

Vse rešitve shranite v eno samo datoteko s končnico `.py` in jo 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 eleganca rešitve.

Dovoljena je uporaba vseh materialov na Učilnici in druge literature na poljubnih medijih. Prepovedana je vsaka komunikacija.

1. Bomboniera (za inspiracijo hvala kolegom s FMF!)

Benjamin je našel bomboniero, v kateri je **širina** stolpcev in **visina** vrstic bombonov. Že pred njim jo je našla in nekoliko izropala njegova sestra Ana. Benjamin se ne bo dotaknil vrstic in stolpcev, iz katerih je Ana že vzela kak bombon. Pojedel pa bo vse ostale. Napiši funkcijo `bomboniera(širina, visina, pojedeno)`, ki pove, koliko bombonov bo pojedel. Seznam `pojedeno` vsebuje pare z vodoravnimi in navpičnimi koordinatami.

Če pokličemo `bomboniera(8, 5, [(2, 1), (2, 4)])` bo Benjamin pojedel vse bombone razen tistih v drugem stolpcu ter tistih v prvi in četrti vrstici, saj je tam stikala že Ana.

2. Vreme

Napiši funkcijo `izpis_vrstice(kraj, vreme, temperatura, veter, tlak)`, ki prejme vremenske podatke: prva dva argumenta sta niza. Ostali argumenti so številke ali pa prazen niz, če podatek ni znan. Funkcija vrne niz z opisom vremena (glej teste). Kraj je izpisan na 35 mest, vreme na 20, temperatura in veter na 5 ter tlak na 8.

Napiši funkcijo `izpisi_vreme(datoteka)`, ki prejme ime datoteke z vremenskimi podatki in vrne niz s celotnim "vremenskim poročilom" (glej teste). Kako je videti datoteka, si oglej v `vreme.txt`, ki se uporablja tudi v testih.

3. Prafaktorji

Napišite funkcijo `prafaktorji(n)`, ki razcepi podano število `n` na prafaktorje in vrne razcep v obliki slovarja. Če pokličemo `prafaktorji(1400)`, vrne slovar `{2: 3, 5: 2, 7: 1}`, saj je $1400 = 2^3 \cdot 5^2 \cdot 7^1$.

Nato napišite funkcijo `gcd(a, b)`, ki prejme dva slovarja, kakršna vrača prejšnja naloga, in vrne največji skupni delitelj števil, ki ju predstavljata tadv slovarja. Če pokličemo `gcd({2: 3, 5: 2, 7: 1}, {2: 2, 7: 2, 11: 1})`, vrne 28. Prvi slovar namreč predstavlja število 1400 in drugi število 2156, njun največji skupni delitelj pa je 28. Da bi dobil vse točke, nalogo reši, ne da bi izračunal števili (npr. 1400 in 2156). Delaj s slovarjema, ki si ju dobil.

Namig: $1400 = 2^3 \cdot 5^2 \cdot 7^1$ in $2156 = 2^2 \cdot 7^2 \cdot 11^1$, zato je njun največji skupni delitelj enak $2^2 \cdot 7^1$.

4. Prafaktorji - rekurzivno

Če želiš (ni pa nujno), si pripravi funkcijo `najm_prafaktor(n)`, ki vrne najmanjše praštevilo, ki deli `n`.

Napiši funkcijo `prafaktorji_rec(n)`, ki vrača isto kot `prafaktorji(n)`. Funkcija **mora biti rekurzivna**: ne sme vsebovati zanke in klicati nobene druge funkcije, razen, če želiš, `najm_prafaktor` (in, če hočeš, `defaultdict`).

Namig: če je `p` praštevilo, ki deli `n`, je razcep enak razcepu števila `n` deljeno s `p`, ki mu dodaš še faktor `p`.

5. Pisni izdelek

Napiši razred `PisniIzdelek`, ki bi bil uporaben za ocenjevanje študentk ali študentov na izpitu.

- Konstruktor prejme in shrani ime študentke oz. študenta.
- `daj_tocke(naloga, tocke)` zabeleži, da je študent pri podani nalogi (naloge so oštevilčene od 1 do 5!) dobil podano število točk. Če funkcijo pokličemo večkrat za isto nalogo, si zapomni zadnje število.
- `rezultat()` vrne rezultat v obliki ("`Janez Novak`", `(20, 10, None, 0, 20)`); prvi element rezultata je ime študenta, drugi pa terka, katere elementi so število točk pri tej nalogi, oz. `None`, če za to nalogo nismo poklicali `daj_tocke`.
- `vsota()` vrne skupno število točk.
- `naredil()` vrne `True`, če je skupno število točk večje ali enako 50; `False`, če ni.
- `ocena()` vrne 5, če je število točk manjše od 50; 6, če je število točk med 50 in 59; 7, če je med 60 in 69, ..., 10, če med 90 in 100. (Opomba: to ni nujno lestvica za ocenjevanje pri tem ali drugih predmetih!)