

Vse rešitve shranite v eno samo datoteko s končnico `.py` in jo oddajte prek Učilnice. Vse funkcije naj imajo takšna imena, kot jih predpisuje naloga. **Pozorno preberite** naloge in ne rešujte le na podlagi primerov!

Da rešitev ne bi imela trivialnih napak, **jo preverite s testi** v ločeni datoteki na Učilnici. Za rešitev naloge lahko dobite določeno število točk, **tudi če ne prestane testov**. Funkcija, ki prestane vse teste, **še ni nujno pravilna**.

Pri reševanju nalog je dovoljena vsa literatura na poljubnih medijih, ves material, ki je objavljen na Učilnici, vključno z objavljenimi programi; njihova uporaba in predelava se ne šteje za prepisovanje.

Izpit morate pisati na fakultetnih računalnikih. Študenti s predolgimi vratovi in podobnimi hibami bodo morali zapustiti izpit. Hujše kršitve, na primer uporabo telefonov in podobno, bomo prijavili disciplinski komisiji.

Študentom **Kognitivne znanosti** je potrebno za 100 % rešiti štiri naloge. Naloga iz snovi, ki se vas ne tiče, je zadnja, vendar jo lahko, če želite, rešite; točke se vam bodo upoštevale.

## 1. Soda in liha

Napišite funkcijo `soda_liha(s)`, ki kot argument prejme seznam števil `s`. Vrstni red števil naj spremeni tako, da se bodo v njem izmenjevala soda in liha števila, začenši s sodim. Oboja naj bodo v enakem vrstnem redu, kot so bila prej. Če je sodih več kot lihah ali obratno, naj odvečna števila odstrani. Funkcija ne sme vračati ničesar.

### Primer

```
>>> s = [5, 8, 4, 2, 8, 17, 13, 10, 2, 4, 6, 8, 9]
>>> soda_liha(s)
>>> s
[8, 5, 4, 17, 2, 13, 8, 9]
```

Po klicu funkcije so soda števila v enakem zaporedju kot prej (8, 4, 2, 8), mednje pa so prepletena liha v enakem zaporedju kot prej (5, 17, 13, 9). Odvečna soda števila (10, 2, 4, 6, 8) so izpuščena.

## 2. Decimalno

Napišite **rekurzivno** funkcijo `decimalno(s)`, ki kot argument prejme niz `s`, ki predstavlja število v dvojiškem zapisu, kot rezultat pa naj vrne to število.

### Primer

```
>>> decimalno('101010')
42
```

**Namig:** k zadnji številki prištej dvokratnik števila, ki ga dobiš, če vse razen zadnje številke pretvoriš iz dvojiškega zapisa v število.

## 3. Ograje

Imamo naselje vrtičkarjev, ki je pravokotne oblike in je razdeljeno na  $n \times m$  kvadratnih polj s stranico 1. Med dvema poljema je ograja, če imata različna lastnika. Primer je narisano na desni. Kdo je lastnik katerega vrtička, je zapisano v seznamu nizov; za primer na desni bi bil seznam takšen: `["AAABC", "ABCD", "ACCD"]`.

A	A	A	B	C
A	B	C	D	C
A	C	C	D	A

Napišite funkcijo `ograje(s)`, ki dobi seznam v tej obliki in vrne skupno dolžino vsej ograji – tako teh med vrtički kot vseh okrog njih.

**Namig:** morda vam bo lažje reševati tako, da najprej izračunate, koliko ograje bi potrebovali, če bi bili vsi lastniki različni. Nato od tega preštejte, koliko je polj, med katerimi je ograja odveč. Nekaj podobnega smo namreč že počeli.

## 4. Tečajji

Univerza v Ljubljani je začela ponujati spletne tečaje. Kdo je že opravil kateri tečaj, zapisujejo v slovar, katerega ključi so imena oseb, pripadajoče vrednosti pa množice tečajev, ki jih je dotična oseba opravila. Če je takšnemu slovarju ime, recimo, `tecaji` in je `tecaji['Ana']` enako `{'Zgodovina gozdarstva', 'Sodobno roparstvo', 'Feminizem na Kočevskem'}`, to pomeni, da je Ana opravila naštetete tečaje. Napisati boste morali več funkcij.

- `opravil(ime, tečaj, tecaji)` naj v slovar `tecaji` pribeleži, da je oseba opravila `tečaj`. (Če je v preteklosti že opravila ta tečaj, naj funkcija ne spremeni ničesar.)
- `najbolj_ucen(tecaji)` naj vrne ime osebe, ki je doslej opravila največ tečajev. Če je takšnih oseb več, naj vrne poljubno med njimi.
- `vsi_tecaji(tecaji)` naj vrne množico imen vseh tečajev, ki se pojavijo v slovarju.
- `neopravljeni(ime, tecaji)` naj vrne imena tečajev, ki jih je že kdo opravil, oseba `ime` pa še ne.

Predpostaviti smete, da je `tecaji` tipa `defaultdict(set)` in ne navaden `dict`.

## 5. Blok

Definirajte razred `Naselje`, ki opisuje stanovanjsko naselje s poimenovanimi hišami. Ima naj naslednje metode.

- Konstruktor sprejme kot argument seznam z imeni hiš.
- `vseli(hisa, ime)` v hišo `hisa` vseli osebo z imenom `ime` in vrne `True`. Če je hiša že zasedena, pusti v njem prejšnjega lastnika in vrne `False`.
- `stanovalec(hisa)` vrne ime stanovalca v podani hiši. Če je še prazna, vrne `None`.
- `izseli(hisa)` sprazni podano hišo. Metoda ne vrne ničesar.
- `prostih()` vrne število prostih hiš.