

1. Podvig

Na sliki so nadmorske višine (v dekametrih) na kvadratni mreži z neenakomernim desnim robom. Višine predstavimo s seznamom različno dolgih seznamov (pod sliko).

Kolesar mora vedno goniti v klanec. V vsaki točki se razgleda v vse štiri smeri ($\wedge$, $\succ$, $\vee$ in $\prec$) in gre v tisto, v katero je klanec največji. Če je klanec enak v več smeri, ima $\wedge$ prednost pred $\succ$, $\succ$ ima prednost pred $\vee$ in $\vee$ pred $\prec$. Če v nobeno smer ni klanca, temveč so vsa sosedna polja enako visoka kot trenutno ali celo nižja, obupano usede na tla, se zjoka in ostane, kjer je.

Napišite funkcijo `podvig(visine, x, y)`, ki prejme seznam višin in začetni koordinati kolesarja ter vrne končni koordinati. Klic `podvig(visine, 0, 5)` vrne `(0, 7)`: šel bo $\succ \vee \vee \prec$. Klic `podvig(visine, 2, 3)` vrne `(1, 3)`: gre $\prec$ in joka.

	0	1	2	3	4	5
0	12	15	32	24	24	
1	8	12	29			
2	3	6	24	13	11	18
3	-2	25	8	10		
4	0	1	3	6	12	16
5	0	2	2			
6	0	13	12	14	16	
7	15	14	5			

```
visine = [
    [12, 15, 32, 24, 24],
    [ 8, 12, 29],
    [ 3,  6, 24, 13, 11, 18],
    [-2, 25,  8, 10],
    [ 0,  1,  3,  9, 12, 16],
    [ 0,  2,  2],
    [ 0, 13, 12, 14, 16],
    [15, 14,  5]]
```

2. Vrh

Kolesar iz drugega primera v prejšnji nalogi bi lahko šel $\wedge \wedge \wedge$ in prispel na 2, 0 z višino 32 (vendar tega ni storil, ker $25 > 24$). Tako bi še vedno v vsakem koraku vozil v klanec (čeprav ne najhujšega) in se zjokal višje (na višini 32, ne 25).

Napišite funkcijo `vrh(visine, x, y)`, ki prejme začetni koordinati in višino ter vrne koordinati in višino najvišjega polja, ki ga kolesar lahko doseže, če vedno vozi navzgor. Klic `vrh(visine, 2, 3)` vrne `(2, 0, 32)`. Če je dosegljivih več enako visokih najvišjih polj, vrne poljubno od njih.

3. Aktivnosti

Tip ni le kolesar. En dan je tek, kolesaril in igral badminton. Drug dan je samo tek. Tretji dan je plaval, kolesaril in se sprehajal ... Aktivnosti si zapisuje v seznam množic; vsaka predstavlja en dan.

Napišite naslednje funkcije.

- `vse_aktivnosti(aktivnosti)` vrne množico vseh aktivnosti, ki se pojavijo v seznamu.
- `naj_veriga(aktivnosti, aktivnost)` vrne najdaljše število zaporednih dni, v katerih se je ukvarjal s podano aktivnostjo. Tako `naj_veriga(aktivnosti, tek)` vrne 3, ker je nekoč tek tri dni zapored.
- `naj_aktivnost(aktivnosti)` vrne ime aktivnosti z najdaljšo verigo in dolžino verige, na primer `( "tek", 3 )`.

```
[{"tek", "kolesarjenje", "badminton"},
 {"tek"},
 {"plavanje", "kolesarjenje", "sprehod"},
 {"tek", "plavanje", "sprehod"},
 {"tek", "kolesarjenje"},
 {"tek", "kolesarjenje"},
 {"plavanje", "sprehod"}]
```

4. Seznam

V resnici si zapiskov ne dela v seznam temveč v datoteko, katere obliko kaže slika na desni. V prvem delu ustreza vsaka vrstica enemu dnevu; črke predstavljajo aktivnosti tega dneva. Sledi vrstica s tremi minusi, tej pa sledijo vrstice, ki povedo, katera črka pomeni katero aktivnost. V različnih datotekah lahko črke pomenijo različne aktivnosti.

Napišite funkcijo `preberi_aktivnosti(ime_datoteke)`, ki prejme ime datoteke v takšni obliki in vrne seznam v obliki iz prejšnje naloge. Primera datotek se pojavita ob prvem zagonu testov.

```
TKB
T
PKS
TPS
TK
TK
PS
---
T=tek
K=kolesarjenje
B=badminton
P=plavanje
S=sprehod
```

5. Spomeniki

Napišite razred `Spomeniki` s primernim konstruktorjem brez argumentov in naslednjimi metodami.

- `obisci(kolesar, spomenik)` zabeleži, da je kolesar obiskal "spomenik", npr. `obisci("Ana", "Vršič")`. Metoda vrne `True`, če je to prvi obisk tega spomenika in `False`, če je bil ta spomenik že kdaj obiskan (s strani tega ali kakega drugega kolesarja).
- `obiskanost(spomenik)` vrne število različnih kolesarjev, ki so obiskali podani spomenik.
- `spomenikov(kolesar)` vrne število različnih spomenikov, ki jih je obiskal podani kolesar.
- `prvaki()` vrne slovar, katerega ključi so imena spomenikov, pripadajoča vrednost pa ime prvega kolesarja, ki je obiskal ta spomenik.

Testi vsebujejo jasen primer.