

II kolokvijum

Na **Desktop**-u u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksa** i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka **mora** da se nalazi u fajlu **Zadatak1.py**. Rešenje 2. zadatka **mora** da se nalazi u fajlu **Zadatak2.py**. Rešenje 3. zadatka **mora** da se nalazi u fajlu **Zadatak3.py**.

Napomena: Ulaz i izlaz se **moraju** učitavati/ispisivati u formatu koji je prikazan u primerima.

1. Pera se igra kartama. Sve karte koje je imao u ruci je složio tako da prvo idu sve karte sa parnim brojevima, a zatim one sa neparnim brojevima (moguće je i da je Pera imao samo parne ili samo neparne karte). Napiši program koji proverava da li je Pera ispravno složio karte.

Ulaz: Sa standardnog ulaza učitavaju se brojevi karata (prirodni brojevi između 2 i 10) sve dok se ne dođe do kraja ulaza (on se može uneti sa 0). Karata ima najmanje dve, a najviše deset.

Izlaz: Na standardni izlaz ispisati "DA" ako je Pera dobro složio karte tj. "NE" u suprotnom.

Ulaz:	Izlaz:
2 6 4 5 3 0	DA
2 6 3 5 4 0	NE

2. Ako pri zapisivanju reči **r** izostavimo jedno slovo onda takav zapis zovemo pogrešno pojavljivanje reči **r**. Na primer, "sea" je pogrešno pojavljivanje reči "seal", zato što je izostavljeno tačno jedno slovo iz početne reči, a redosled ostalih slova je isti.

- a) Napisati funkciju koja proverava da li je prosleđena reč pogrešno pojavljivanje reči **r**.
- b) Napisati funkciju koja u stringu **s** ispravlja sva pogrešna pojavljivanja reči **r**.

Ulaz: Na standardnom ulazu se u prvom redu unosi reč **r**, a zatim u drugom redu string **s**.

Izlaz: Ukoliko u stringu **s** postoje reči koje predstavljaju pogrešno pojavljivanje reči **r**, ispisati "DA" i u narednom redu ispravljen string **s**. Ukoliko u stringu **s** ne postoje takve reči, ispisati "NE".

Ulaz:	Izlaz:
seal can you see the sea	DA can you see the seal
seal can you see the seals	NE

3. Planinska ekspedicija se kretala pravougaonim terenom i u pravilnim intervalima merila je nadmorsku visinu. Pretpostavljamo da su prikupljene visine u matrici dimenzija **n** x **m**. Kada se nalaze u nekoj tački, planinari mogu da gledaju na istok, zapad, sever i jug. U svakom od ta 4 smeru planinari mogu da vide nekoliko vrhova. Tačka se smatra vidljivom, ako se ispred nje nalaze samo tačke koje imaju strogo manju nadmorsku visinu od nje (ili ako ispred nje nema drugih tačaka).

Ulaz: U prvom redu se unosi broj vrsta **n**, u drugom redu broj kolona **m**. Zatim se u narednih **n** x **m** redova unose brojevi koji predstavljaju nadmorske visine vrhova. U pretposlednjem redu unosi se broj vrste **v**, a u poslednjem broj kolone **k** tačke na kojoj se nalaze planinari.

Izlaz: Broj vrhova vidljivih iz tačke na kojoj se nalaze planinari.

Ulaz:	Izlaz:
4	6
5	
3	
8	
5	
4	
2	
4	
1	
7	
6	
5	
2	
1	
4	
8	
9	
3	
3	
7	
5	
2	
1	
2	

Dodatno objašnjenje:

Ulaz:

4 5

3 8 5 4 2

4 1 7 6 5

2 1 4 8 9

3 3 7 5 2

1 2

Izlaz:

6

Planinar iz tačke sa koordinata (1,2) koja je na visini 7, u pravcu ka zapadu vidi tačke sa nadmorskim visinama 1 i 4, u pravcu istoka samo tačku sa nadmorskom visinom 6, u pravcu ka severu vidi tačku sa nadmorskom visinom 5, u pravcu ka jugu tačke sa visinama 4 i 7. Dakle, ukupno vidi 6 vrhova.