**PROG1 : Projekat**

**(Covece ne ljuti se)**

**Cilj projekta**

Cilj projekta je uraditi simulaciju sa 2 igraca koji igraju Covece ne ljuti se.

Pravila igre mozete naci na:

<https://sk.wikipedia.org/wiki/%C4%8Clove%C4%8De_nehnevaj_sa>

Pravila igre na englenskom jeziku:

<https://en.wikipedia.org/wiki/Mensch_%C3%A4rgere_Dich_nicht>

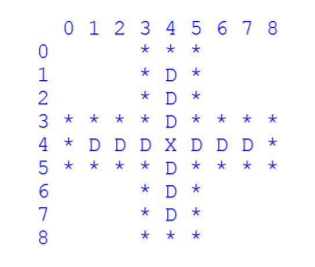
Postoji vise verzia pravila. Ako se i malo odaljite od pravila koje navodi Wikipedia, nece biti tako velik problem.

**Prvi deo (5 bodova)**

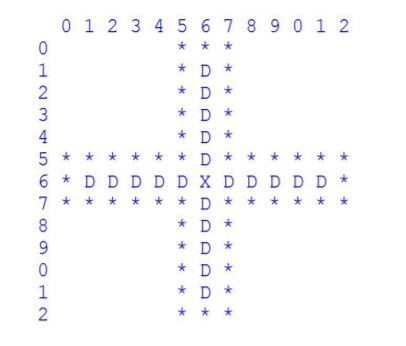
Naprogramirajte funkciju *gensachovnica* sa parametrom n, koja ce generisati i vratiti „sahovsku tablu“ dimenzija nxn za *Covece ne ljuti se*. Onda naprogramirajte funkciju *tlacsachovnicu* koja ce imati argument sachovnicu a koja ce tu sahovsku tablu nacrtati (ispisati).

Tabla ima oblik krsta gde sirina ramena je 3. Broj n (t.j. dimenzije table) mora da bude neparan.

Naprimer za n = 9 izgledace ovako:

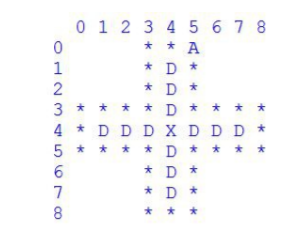


Za n = 13 izgledace tabla ovako:



**Drugi deo (8 bodova)**

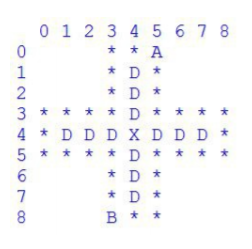
Simulirajte kretanje jednog pijuna po tabli. Korisnik ce moci napocetku da izabere velicinu table. Napocetku igre pijun ce se nalaziti u gornjem delu ramena kao na slici:



Sledece je da se vigenerisu nasumicno brojevi od 1 do 6. Za generisanje brojeva koristite funkciju *randint* iz biblioteke random (<https://docs.python.org/3.8/library/random.html>). Na osnovu toga, koji broj ce se vigenerisati, pijun ce se pomeriti o odredeni broj bolja. Onda ce se opet vigenerisati novi grojevi (nasumicno) i tako dalje. Simulacija se zavrsava, kada ce se pijun nalaziti u kucici (na jednu iz pozicia oznacenih slovom D) u gornjem delu ramena. Simulacija mora da ispise svaki nasumicno vigenerisan broj a onda po vigenerisanom novom broju simulacija mora da nacrta tablu sa aktualnom pozicijom pijuna.

**Treti deo (7 bodova)**

Simulirajte igru sa 2 igraca A,B. Korisnik ce moci napocetku igre odabrati velicinu table. Startna pozicija igraca A je u gornjem uglu ramena, dok startna pozicija igraca B je u donjem ramenu (pogledaj sliku nize):



Svaki igrac ima (n-3)/2 pijuna. Pobeduje onaj igrac koji kao prvi uspe da umesti sve svoje figure u kucicu. Igrac A se trudi da smesti svoje figure u kucicu u gornjem delu ramena, a igrac B se trudi da svoje figure smesti u kucicu u donjem delu. Vasa smulacija mora da ispise svaki vigenerisan broj a po svakom novom vigenerisanom broju mora simulacija uvek da nacrta tablu sa aktualnom poziciom figura.

Naprimer vasa asimulacija moze funkcionisati na sledeci nacin, igrac moze pomerati samo jednu figuru sve dok ne dode do kucice , a tek onda ce postaviti novog pijuna na startnu poziciju. To znaci da ce svaki igrac u tom trenutku imati samo jednu figuru napolju.

Vasa simulacija moze raditi i na nacin da ce igrac imati vise figura na polju u isto vreme. U tom slucaju mozete da izaberete neki nacin (pravilo),a na osnovu njega ce maci da igrac bira koju figuru da pomeri. (naprimer: igrac moze da pomeri tu figuru koj ase nalazii blize kucici)

***Upozorenje:***

U ovom zadatku morate napisati program, koji korisniku omugucuje da pusti simulaciju igre *Covece ne ljuti se* sa dva igraca. Korisnik moze napocetku da izabere velicinu table a onda treba da krene simulacija. Simulacija bi trebala da se zavrsi kada jedan igrac pobedi, a nebi trebala od korisnika da trazi nikakve druge informacije (inputs). Kada korisnik pusti ovu simulaciju videce u konzoli mnogo ispisanih (nacrtanih) tabli a nakraju ce saznati koji igrac je pobedio, bez toga da bi korisnik mogao nekako da utice na tok igre.

Ako zelite, mozete da umesto ovakve simulacije uradite program, koji ce korisniku omuguciti da „igra“ *Covece ne ljuti se* za dva igraca. Ovakav program bi od korisnika trazio da izabere, koju figuru treba da pomeri.