

1. Napraviti vlastiti paket koji radi testiranje performansi računalnih programa. Performanse će se vrednovati mjerenjem duljine trajanja izvođenja funkcije. Kod koji testira tvoj paket dostupan je na službenom repozitoriju tečaja, pod nazivom `ilpc-dz02.py`; ovu datoteku potrebno je preuzeti i poželjno je proizvoljno mjenjati.

Ime paketa neka bude `test`. Neka iz paketa bude "importabilna" (u nedostatku bolje riječi) samo jedna funkcija, koja se isto zove `test`. Funkcija treba imati iduću signaturu:

```
def test(function, arguments, repeat_count)
```

gdje je `function` funkcija koja se poziva, `arguments` tuple argumenta s kojima se funkcija poziva, a `repeat_count` broj puta kojim će se mjerenje ponoviti. Povratna vrijednost funkcije su razne statistike: apsolutni minimum za implementaciju je prosjek. Varijanca je poželjna ali ne i nužna. Dajem bonus bodove za one koji naprave extra stvari poput 95% intervale za procjenitelje i za vrijednost populacije, ali to samo jer volim statistiku :))

Nadograditi `ilpc-dz02.py` tako da testira nasumične cijele brojeve iz intervala `range()` objekta. ** S obzirom da još nismo učili kako pozvati funkciju s arbitrarnim brojem argumenata, dovoljno je prilagoditi paket `test` samo ovoj zadaći, odnosno, možete pretpostaviti da će broj argumenata funkciji uvijek biti 4. Naravno, oni koji naprave drukčije dobe bonus bodove.

2. Nakon što se uvjeriš da ti paket radi kako treba, napravi CLI alat sa `argparse` modulom. Cilj je da se `ilpc-dz02.py` može pokrenuti kao skripta s argumentima. Argumenti su ono što ti odlučiš da želiš da budu argumenti, ali pravi se da želiš ovo koristiti u praksi, stoga je u redu poslati argumente poput `start/stop/step/repeat_count`, ali i druge poput statistika koje računaš.
3. Program gleda nalaze li se brojevi (nasumični ili ne, ovisno o implementaciji) u rasponu brojeva, odnosno `range()` objektu. Istraži kako složenost ispitivanja raste s "veličinom" `range()` objekta, odnosno dolazi li do usporavanja/ubrzanja prilikom mjenjanja parametara `start/stop/step`. Naravno, pazi da gledaš prosječno vrijeme *po petlji*. Nakon što doneseš *data backed* zaključke, napiši mi ih u mailu prilikom predaje :D

Napomena 1: cijenim ako napraviš svoju implementaciju računanja statističkih momenata, ali to nije nužno. Ako želiš, istraži paket iz PSL koji to čini za tebe.

Napomena 2: kod treba biti čitljiv djetetu iz trećeg osnovne sa osnovnim znanjem engleskog jezika. Tvoj kod možda, a možda i ne, bude provjeren sa djetetom iz trećeg osnovne.