Uvod

Dorotea puno i često putuje. Kako su duža putovanja nerijetko dosadna, Dorotea je smislila mentalnu igru uz koju joj vrijeme brže prolazi. Gledajući aute koje prolaze, Dorotea pokušava pronaći registarske tablice za koje vrijedi slijedeće:

*“Koristeći matematičke operacije i početne 3 znamenke registracijskog broja dobiti posljednji broj registracije”*

Iako je Dorotea već dosta iskusna u igri, zanima ju malo detaljnije statistike u vezi igre koju je smislila.

Zadatak 1.

Napraviti Python skriptu koja će zapisati popis svih registracijskih oznaka, broj mogućih rješenja te jedan primjer rješenja (ako postoji) u .csv datoteku.

Skripta prilikom izvođenja mora za svaki registracijski broj izbrojati koliko različitih matematičkih jednadžbi se može postaviti da vrijedi uvjet igre.

Matematičke operacije koje treba implementirati

1. Zbrajanje

2. Oduzimanje

3. Množenje

4. Dijeljenje

5. Faktorijel

Izlazni format podataka mora biti csv. Zaglavlje se mora sastojati od sljedećih polja:

1. plate\_num – Broj registracije. Prilikom upisa sa znakom „-„ odvojiti zadnju znamenku od ostalih (primjer ispisa je ispod). Za potrebe rješavanja zadatka uzeti u obzir samo 4-znamenkaste registracije.

2. solution – Ako rješenje postoji, ispisati jednadžbu. U slučaju više rješenja, odabrati i ispisati bilo koju. Tekstualni prikaz rješenja za matematičke operacije neka bude sljedeći:

a. Zbrajanje – a + b

b. Oduzimanje – a - b

c. Množenje – a \* b

d. Dijeljenje – a / b

e. Faktorijel - a!

3. total\_num – Ukupan broj mogućih rješenja igre za navedeni registracijski broj

**Napomene**:

• Uzeti u obzir zagrade (redoslijed računanja) te prikazati ih ispravno u polju solution. • Svaka znamenka može biti pomnožena sa -1 odnosno poprimiti negativnu vrijednost o Npr. Jedno od rješenja igre za registracijski broj 5182 je:

▪ -5 - 1 + 8 = 2 odnosno 5\*(-1) - 1 + 8

• Pripaziti na rubne slučajeve, ignorirati računanje vrlo velikih brojeva

o npr. 9!9! može napraviti probleme u sustavu

Primjer formata izlazne .csv datoteke je sljedeći:

| **plate\_num,solution,total\_num**  000-3,(0! + 0!) + 0! = 3,2  000-4,,0  …  533-3,5 - (3! / 3) = 3,5  534-0,,0  … |
| --- |

*registration.csv*

**Bonus pitanje**:

Koje promjene u kodu treba napraviti ako želimo samo popis tablica i zastavicu (primjer ispisa ispod) koja prikazuje je li moguće doći do zadnje znamenke preko uvjeta igre? Koje bi sve modifikacije u kodu poboljšale performanse skripte da se izvede u najkraćem mogućem vremenu?

| **plate\_num,can\_be\_solved**  000-3,true  000-4,false  …  533-3,true  534-0,false  … |
| --- |

Zadatak 2.

Koristeći registration.csv datoteku stvoriti dashboard unutar Splunka koji će prikazivati osnovnu statistiku o igri (slika ispod):



Napomena: Ovisno o opsegu Vašeg rješenja, podaci na vašem dashboardu neće nužno biti isti kao sa slike iznad.

Koristeći *inputlookup* naredbu, potrebno je napisati Splunk searcheve za sljedeće statistike:

• Broj tablica za koje nema mogućih rješenja

• Broj tablica za koje postoji barem jedno rješenje

• Postotak tablica za koje postoji barem jedno rješenje

• Tablica za koju ima najviše rješenja a da zadnja znamenka nije trivijalna (0 ili 1) • Popis tablica za koje ima točno jedno rješenje

o Ispisati broj tablice i navedeno rješenje

Dostava rješenja

Rješenje treba sadržavati zapakiranu Splunk aplikaciju koja unutar sebe sadrži Python skriptu iz prvog zadatka te xml dashboard iz drugog zadatka. Rješenje bonus pitanja se može dostaviti u tekstualnoj datoteci uz zapakiranu aplikaciju.

Ocjenjivanje rješenja

Kod ocjenjivanja rješenja ćemo posebno obratiti pozornost na ispravnost, urednost te performanse Python koda i Splunk searcheva. Komentari kompliciranijih dijelova programskog koda su dobrodošli.

Korisni linkovi:

https://www.splunk.com/en\_us/download/splunk-enterprise.html

https://docs.splunk.com/Documentation/Splunk/latest/SearchTutorial/WelcometotheSearchTutorial https://www.splunk.com/en\_us/resources/videos/basic-search-in-splunk-enterprise.html

Sretno