

# Igra faktorizacije (igra)

Igra faktorizacije je u posljednje vrijeme postala veoma popularna, čak i kod onih malo starijih. Osmišljena je tako da od svakog igrača izvuče maksimum njegovih matematičkih i logičkih sposobnosti. Pravila su jednostavna. Na početku igre je dat niz cijelih brojeva  $P$  od kojih je potrebno formirati novi niz  $Q$ . Članovi niza  $Q$  nastaju množenjem bilo koja dva člana niza  $P$ , pri čemu se svaki član može množiti i sam sa sobom. Niz  $Q$  sadrži isključivo različite elemente, dakle, ne smije sadržavati dva ista elementa. Nakon što su upotrijebljene sve kombinacije i niz  $Q$  je u potpunosti formiran, potrebno je pronaći  $N$ -ti najmanji član niza  $Q$ . Pobijedio je onaj igrač koji prvi pronađe tačno rješenje. Ukoliko neki od igrača ponudi netačno rješenje, ispada. Gospodin Brko je odlučio napraviti turnir za igru faktorizacije i shvatio je da treba malu pomoć, tačnije, program koji će za zadani niz  $P$  i zadani broj  $N$  utvrditi tačno rješenje.

## Zadatak

Vaš zadatak je da napravite funkciju *IgraFaktorizacije* koja prima tri parametra. Prvi parametar je cijeli broj  $M$ , drugi parametar je niz cijelih brojeva  $P$  veličine  $M$ , a treći parametar je cijeli broj  $N$ . Funkcija kao rezultat treba da vrati cijeli broj koji predstavlja  $N$ -ti najmanji član niza  $Q$ , pri čemu je niz  $Q$  formiran međusobnim množenjem članova niza  $P$ .

## Primjeri

### *Primjer br. 1*

$IgraFaktorizacije(3, \{7, 2, 5\}, 4) = 25$

*Objašnjenje:* Po završetku množenja, niz  $Q$  sadrži sljedeće članove: 4, 10, 14, 25, 35 i 49. Kao što se vidi, četvrti najmanji član niza  $Q$  je broj 25.

### *Primjer br. 2*

$IgraFaktorizacije(5, \{10, 10, 10, 10, 10\}, 1) = 100$

*Objašnjenje:* Niz  $Q$  sadrži samo jedan član, broj 100.

## Ograničenja

$$1 \leq M \leq 5.000$$

$$1 \leq P_i \leq 100$$

$$1 \leq N \leq \text{veličina niza } Q$$

Vremensko i memorijsko ograničenje, kao i raspored bodova po testnim primjerima su dostupni na sistemu za ocjenjivanje.

## Detalji implementacije

Sa servera za takmičenje možete preuzeti pripremljena okruženja (*igra\_c.zip*, *igra\_cpp.zip* ili *igra\_pas.zip*) sa osnovnim fajlovima za C, C++ i Pascal.

Ukoliko koristite C ili C++, napišite funkciju s prototipom:

**int** IgraFaktorizacije (**int** M, **int\*** P, **int** N); u fajlu *igra.[c/cpp]*.

Ukoliko koristite Pascal, napišite funkciju sa prototipom:

**function** IgraFaktorizacije(M : **LongInt**; P : **Array of LongInt**; N : **LongInt**) : **LongInt**; u fajlu *igra.pas*.

Samo unutar ovog fajla treba da implementirate svoje rješenje. Pri tome smijete koristiti i druge pomodne funkcije koje ste vi napisali, te standardna zaglavlja/biblioteke odabranog programskog jezika i funkcije iz ovih biblioteka. Ne smijete ni na koji način vršiti interakciju sa standardnim ulazom/izlazom niti sa bilo kojom datotekom.

U pripremljenom okruženju nalazi se fajl *grader.[c/cpp/pas]* koji testira ispravnost rada<sup>1</sup> funkcije koju ste napisali na javne testne primjere, dok komisija koristi svoj *grader.[c/cpp/pas]* koji nije javni. U skladu s tim, slobodni ste da modifikirate *grader.[c/cpp/pas]* i prilagođavate ga svojim potrebama u svrhu testiranja na lokalnom računaru. Kada šaljete svoje rješenje, šaljete samo fajl *igra.[c/cpp/pas]*. Ukoliko koristite *Code::Blocks* u pripremljenim okruženjima možete naći i odgovarajuće projekte sa podešenim parametrima za prevođenje. *Release build* u potpunosti odgovara parametrima za prevođenje koji su na serveru za takmičenje, dok *Debug build* ima isključene optimizacije i uključene simbole za debugiranje. Ukoliko koristite *FreePascal IDE*, dovoljno je da pokrenete prevođenje fajla *grader.pas* dok je u istom folderu fajl *igra.pas*. Na serveru za takmičenje postavljeni su sljedeći parametri za prevođenje: -dEVAL -vw -XS -O2. Ukoliko ne koristite *Code::Blocks*, odnosno *FreePascal IDE*, u okruženjima se nalaze i fajlovi *prevedi\_[c/cpp/pas].sh* koje možete koristiti za prevođenje svojih programa, a koje pozivate iz terminala komandom *sh prevedi\_[c/cpp/pas].sh* iz odgovarajućeg foldera.

---

<sup>1</sup> Fajl *grader.[c/cpp/pas]* koji je javno dostupan testira samo ispravnost bez postavljanja ograničenja na vrijeme izvršavanja i iskorištenu memoriju.