

Loto (loto)

Broj kombinacija na igri lotu se računa na sljedeći način:

$$C = \frac{N!}{(N-M)! \cdot M!}$$

Operacija $N!$ predstavlja faktorijel, i računa se na sljedeći način:

$$0! = 1;$$

$$N! = 1 \cdot 2 \cdot \dots \cdot N; (N > 0)$$

Na primjer, za loto 5/39 gdje izvlačimo 5 kuglica od 39 mogućih, taj broj je: 575757. Svemirska Lutrija d.o.o. želi da organizuje igru Loto za našu galaksiju. Da bi odredili fond dobitaka i cijenu listića, moraju znati koliki je broj kombinacija za igru Lota koju planiraju. Rukovodioci Svemirske lutrije znaju kako je to veoma težak proračun, pa bi se zadovoljili i rješenjem koje predstavlja sumu cifara tog broja.

Zadatak

Vaš zadatak je da napravite funkciju *SumaCifara* koja prima dva parametra, cijeli broj N i cijeli broj M koji predstavljaju ukupan broj kuglica i broj kuglica koje se izvlače, respektivno. Funkcija vraća cijeli broj koji predstavlja sumu cifara broja loto kombinacija za date brojeve N i M .

Primjeri

Primjer br. 1

$$\text{SumaCifara}(10, 5) = 9$$

Objašnjenje: Broj kombinacija za brojeve 10 i 5 je 252. Suma cifara broja 252 je 9.

Primjer br. 2

$$\text{SumaCifara}(20, 5) = 15$$

Objašnjenje: Broj kombinacija za brojeve 20 i 5 je 15504. Suma cifara broja 15504 je 15.

Ograničenja

$$1 \leq M \leq N \leq 1000$$

Vodite računa da je 1000! broj sa 2568 cifara.

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 (*loto_c.zip*, *loto_cpp.zip* ili *loto_pas.zip*) sa osnovnim fajlovima za *C*, *C++* i *Pascal*.

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

int SumaCifara(**int** N, **int** M); u fajlu *loto.[c/cpp]*.

Ukoliko koristite *Pascal*, napišite funkciju sa prototipom:

function SumaCifara(N : **LongInt**; M : **LongInt**) : **LongInt**; u fajlu *loto.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 rada¹ 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 *loto.[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 *loto.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.

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