

Zlato na Marsu (mars)

Nedavno je na Marsu otkriveno polje sa zlatnom rudom. Iz praktičnih razloga, to polje je aproksimirano matricom dimenzija $N \cdot M$, gdje je N broj redova, a M broj kolona matrice. Svakoj ćeliji matrice odgovara jedan kvadratni metar polja, odnosno, ukupna vrijednost zlata na tom kvadratnom metru polja. Za prikupljanje zlata sa površine, koristi se bespilotna letjelica AZ-125. Kada sleti na određeni dio polja, zbog tehnoloških ograničenja letjelica zlato može prikupljati samo sa svoje lijeve i desne strane, ispred i iza sebe, i naravno s polja na kojem se nalazi. Odnosno, ukoliko bismo to predstavili pomoću prethodno spomenute matrice, letjelica može prikupljati zlato samo iz onog reda i one kolone u kojima se nalazi ćelija matrice koja predstavlja mjesto na koje je letjelica sletila. Kako proces slijetanja i uzlijetanja troši velike količine goriva, letjelica zlato kupi samo jednom.

Zadatak

Vaš zadatak je da napišete funkciju *MaksimalnaVrijednost* koja prima tri argumenta, cijele brojeve N i M , kao i niz cijelih brojeva T veličine $N \cdot M$ koji sadrži sve vrijednosti zlata spomenute matrice, na način da prvih M elemenata niza čini prvi red, narednih M elemenata čini drugi red, itd. Funkcija vraća cijeli broj koji predstavlja maksimalnu vrijednost zlata koju letjelica AZ-125 može prikupiti.

Primjer

$\text{MaksimalnaVrijednost}(5, 5, \{5, 6, 7, 7, 7, 5, 6, 8, 8, 8, 5, 6, 9, 9, 9, 5, 6, 11, 11, 11, 5, 6, 16, 15, 12\}) = 89$

5	6	7	7	7
5	6	8	8	8
5	6	9	9	9
5	6	11	11	11
5	6	16	15	12

Slika 1 – Prikaz rješenja datog primjera

Ograničenja

$$1 \leq N \leq 5.000$$

$$1 \leq M \leq 5.000$$

$$1 \leq T_i \leq 10.000$$

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 (*mars_c.zip*, *mars_cpp.zip* ili *mars_pas.zip*) sa osnovnim fajlovima za C, C++ i Pascal.

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

int MaksimalnaVrijednost(**int** N, **int** M, **int*** T); u fajlu *mars.[c/cpp]*.

Ukoliko koristite Pascal, napišite funkciju sa prototipom:

function MaksimalnaVrijednost (N : **LongInt**; M : **LongInt**; T : **Array of LongInt**) : **LongInt**; u fajlu *mars.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 modificirate *grader.[c/cpp/pas]* i prilagođavate ga svojim potrebama u svrhu testiranja na lokalnom računaru. Kada šaljete svoje rješenje, šaljete samo fajl *mars.[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 *mars.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.