

# Opasni rudnici (rudnici)

Stanovnici Rudnikostanije, male planete koja se nalazi nedaleko od Zemlje i Sunčevog sistema odlučili su da iskopavaju tzv. opasne rudnike. Problem sa opasnim rudnicima je taj što se oni veoma lako urušavaju i predstavljaju veliki rizik za sigurnost rudara, a smiju se iskopavati samo jednom. Rudnikostanci žele izračunati kolika bi bila maksimalna zarada od iskopavanja nekog opasnog rudnika, pa tek onda odlučiti da li vrijedi rizikovati ili ne. To, međutim, ne znaju uraditi sami, pa su zamolili vas da im pomognete. Dostavili su vam dvodimenzionalne snimke svojih opasnih rudnika, kao npr. ovu sljedeću:

```
      A
    B   C
  D   E   F
G   H   I   J
```

Opasni rudnici su piramidalnog oblika - u prvom sloju rudnika nalazi se jedna ćelija, u drugom sloju dvije ćelije, ..., na  $k$ -tom sloju je  $k$  ćelija. Ćelije sadrže različite rude, slabijeg i boljeg kvaliteta. Svako slovo prikazano na snimku označava cijenu rude koja se nalazi u toj ćeliji rudnika, i to tako što ASCII vrijednost slova predstavlja tržišnu vrijednost specifične rude. Sve oznake su isključivo velika slova engleskog alfabeta, tako da ruda označena slovom A (ASCII=65) vrijedi najmanje, a ruda označena slovom Z (ASCII=90) najviše. Između njih se po vrijednosti nalaze i ostali tipovi ruda. Jedini relativno siguran način za iskopavanje opasnih rudnika je od samog vrha do samog dna, isključivo prema dole i to na lijevu ili desnu stranu. Na konkretnom snimku datom kao primjer, sigurni smjerovi iskopavanja su sljedeći:

- iz A prema B ili C
- iz B prema D ili E
- iz C prema E ili F
- iz D prema G ili H
- iz E prema H ili I
- iz F prema I ili J.

Naučnici koji su vršili snimanje rudnika, vrlo su pragmatični i opis rudnika vam daju u obliku stringa u kojem su redom upisana slova od prvog sloja na niže i na svakom sloju s lijeva na desno. Tako je prethodni snimak rudnika opisan stringom: "ABCDEFGHJIJ". String je uvijek takve dužine da su popunjene sve ćelije do posljednjeg snimljenog sloja.

## Zadatak

Zadatak je napisati funkciju *MaksimalnaZarada* koja kao argumente prima prirodan broj  $n$  i string (maksimalne dužine  $n$  karaktera) koji je sastavljen isključivo od velikih slova engleskog alfabeta. Ovaj string predstavlja ranije opisani opasni rudnik. Funkcija treba da vrati cijeli broj koji predstavlja maksimalanu zaradu prilikom iskopavanja nekog opasnog rudnika pridržavajući se prethodno opisanih pravila sigurnog iskopavanja.

**Podzadatak 1 (12 bodova):**  $1 \leq n \leq 55$

**Podzadatak 2 (13 bodova):**  $1 \leq n \leq 4.095$

**Podzadatak 3 (13 bodova):**  $1 \leq n \leq 20.100$

**Podzadatak 4 (62 boda):**  $1 \leq n \leq 12.502.500$

## Primjeri

### Primjer 1

*MaksimalnaZarada*(6,"CBCDEC") = 203

*Obrazloženje.* Iz proslijeđenog stringa, može se zaključiti da je snimak rudnika kako slijedi.

```
      C
     B  C
    D   E   C
```

Zarada je najveća ako se iskopavanje vrši po putanji: C-C-E. ASCII vrijednost od C je 67, a od E 69, pa je zarada  $67 + 67 + 69 = 203$ .

## Primjer 2

*MaksimalnaZarada*(10,"AGCDECZIZU") = 295

*Obrazloženje.* Iz proslijeđenog stringa, može se zaključiti da je snimak rudnika kako slijedi.

```
      A
     G  C
    D   E   C
   Z    I    Z    U
```

Zarada je najveća ako se iskopavanje vrši po putanji: A-G-E-Z. ASCII vrijednost od A je 65, od G 71, od E 69 i od Z 90, pa je zarada  $65 + 71 + 69 + 90 = 295$ .

## Detalji implementacije

Sa servera za takmičenje možete preuzeti pripremljena okruženja (`rudnici_c.zip`, `rudnici_cpp.zip` ili `rudnici_pas.zip`) sa osnovnim fajlovima za C/C++ i Pascal.

Ukoliko koristite C ili C++ napišite funkciju sa prototipom

```
int MaksimalnaZarada(int n, const char *rudnik);
```

u fajlu `rudnici.[c/cpp]`.

Ukoliko koristite Pascal napišite funkciju sa prototipom

```
function MaksimalnaZarada(n : LongInt; rudnik : String) : LongInt;
```

u fajlu `rudnici.pas`.

Samo unutar ovog fajla treba da implementirate svoje rješenje. Pri tome smijete koristiti i druge pomoćne 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. Kada šaljete svoje rješenje, šaljete samo fajl `rudnici.[c/cpp/pas]`, dok komisija koristi svoj `grader.[c/cpp/pas]` koji nije javni. U skladu s tim, slobodni ste da modifikujete `grader.[c/cpp/pas]` i prilagođavate ga svojim potrebama u svrhu testiranja na lokalnom računaru.

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 prevođenja fajla `grader.pas` dok je u istom folderu fajl `rudnici.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 programe, 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.