

Implementirajte strukturu `Stand`, koja predstavlja štand na nekoj tržnici, i ima konstruktor `Stand(string namirnica, int cijena)`, gdje prvi argument predstavlja namirnicu koja se prodaje na štandu, a drugi cijenu te namirnice u Kn po jedinici mjere (kg, komad, ili što već, ovisno o namirnici). Smijete pretpostaviti da je namirnica neprazan string, a cijena pozitivna za "normalne" štandove. Struktura `Stand` ima metodu `ispisi()`, koja ispisuje namirnicu, razmak, te cijenu i "Kn" unutar uglatih zagrada (bez prelaska u novi red). Primjer:

```
kupus [8Kn]
```

Također implementirajte strukturu `Trznica`, koja predstavlja tržnicu s mrežom od  $m \times n$  (smijete pretpostaviti da su  $m$  i  $n$  između 1 i 30) pozicija na kojima mogu biti locirani štandovi. Tržnica mora imati sljedeće metode:

- konstruktor `Trznica(int m, int n)`, koji stvara tržnicu s recima numeriranim brojevima od 0 do  $m-1$  i stupcima numeriranim brojevima od 0 do  $n-1$ , koja je inicijalno prazna (bez štandova).
- `int dodaj(Stand S, int i, int j)`, koja dodaje štand `S` na poziciju  $i$  u  $j$ -tom stupcu  $i$ -tog retka tržnice. Na istoj poziciji može biti više štandova, ali ne više od 4 (ako na mjestu  $(i,j)$  već postoje 4 štanda, ili su koordinate izvan tržnice, `S` se ne dodaje). Vraća stari broj štandova na toj poziciji (0 ako je pozicija bila prazna ili izvan tržnice, 4 ako je bila puna i `S` nije dodan, 3 ako je pozicija puna nakon dodavanja `S`).
- `int prazno()`, koja vraća broj praznih pozicija na tržnici (inicijalno  $m \times n$ ).
- `double putemTrosi(int p)`, koja vraća iznos koji će potrošiti čovjek koji prođe prolazom  $p$  (koji razdvaja  $(p-1)$ . i  $p$ . redak u tržnici, odnosno prolazi uz 0. redak za  $p=0$ , ili uz  $(m-1)$ . redak za  $p=m$ ) kroz tržnicu, i na svakom štandu pored kojeg prođe kupi jednu jedinicu proizvoda koji se na tom štandu prodaje. Ako je  $p$  manji od 0 ili veći od  $m$ , vraća 0.
- `Stand najjeftiniji(string namirnica)`, koja vraća štand na tržnici na kojem je tražena namirnica najjeftinija (bilo koji, jer svi su jednaki). Smijete pretpostaviti da je parametar neprazan string. Ako nema štanda s traženom namirnicom na tržnici, vraća se "nepostojeći" `Stand("", 0)`.
- `double najblizi(double x, double y)`, koja vraća udaljenost do štanda tržnice najbližeg danim koordinatama (koje ne moraju biti cijele, i mogu biti i izvan tržnice -- smatramo da je tržnica nacrtana kao matrica u IV. kvadrantu koordinatnog sustava, tako da su reci posloženi duž negativnog smjera ordinate, a stupci duž pozitivnog smjera apscise). Ako je tržnica prazna, vraća -1.

Primjer klijentskog programa i izlaz koji taj program proizvede:

```
#include "trznica.h"
#include <iostream>
using namespace std;
int main(){
    Stand Barica("sir",10),Jela("jabuka",7);
    Barica.ispisi();
    Trznica Dolac(5,8);
    cout<<endl<<Dolac.dodaj(Jela,3,4)<<'\t';
    cout<<Dolac.dodaj(Jela,4,7)<<'\t';
    cout<<Dolac.dodaj(Barica,3,4)<<'\t';
    cout<<Dolac.dodaj(Barica,0,0)<<endl;
    for(int p=-2;p<20;++p)cout<<Dolac.putemTrosi(p)<<' ';
    cout<<endl<<Dolac.prazno()<<endl;
    Dolac.najjeftiniji("sir").ispisi();cout<<endl;
    Dolac.najjeftiniji("kupus").ispisi();cout<<endl;
    cout<<Dolac.najblizi(4,-7)<<endl;
    cout<<Dolac.najblizi(0,0)<<endl;
}
```

```
sir [10Kn]
0      0      1      0
0 0 10 10 0 17 24 7 0 0 0 0 0 0 0 0 0 0 0 0
37
sir [10Kn]
[0Kn]
4
0
```