

#define DB 5

struct munka{

string elvegzo;

string azonosito;

int kezdes;

int hossz;

};

bool utkozes(struct munka m1, struct munka m2){

int bef1;

bef1 = m1.kezdes + m1.hossz;

int bef2;

bef2 = m2.kezdes + m2.hossz;

if(bef1 > m2.kezdes and bef2 > m1.kezdes) return true;

else return false;

}

void kiiratas(struct munka m){

cout << m.elvegzo << " " << m.azonosito << " "

<< m.kezdes << " " << m.kezdes << ':' << m.hossz << endl;

}

void rendez(struct munka m[], int n) {

for(int i = n-1; i >= 1; i-- ) {

for(int k = 0; k < i; k++) {

if(m[k].elvegzo > m[k+1].elvegzo) {

munka csere = m[k];

m[k] = m[k+1];

m[k+1] = csere;

}

}

}

}



#include <string>

string szokozkiszed(const string &s){

int hosssz = s.length();

string vissza;

for(int i = 0; i < hosssz; i++){

if (s[i] != ' ') {

vissza += s[i];

}

}

return vissza;

}

void betubolszam(const string &s){

int hosssz = s.length();

for(int i = 0; i < hosssz; i++){

if(s[i] >= 'a' && s[i] <= 'i')

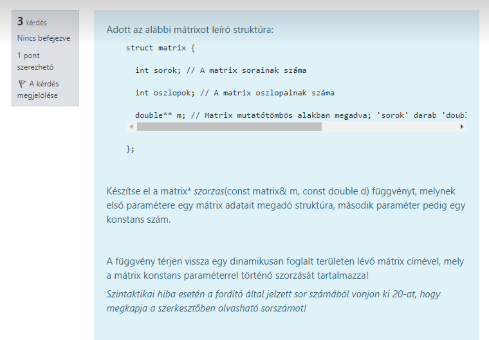
cout << char(s[i] - 'a' + '1');

else

cout << s[i];

}

}



matrix\* szorzas(const matrix& m, const double d) {

matrix\* mtx = new matrix;

mtx->sorok = m.sorok;

mtx->oszlopok = m.oszlopok;

mtx->m = new double\*[mtx->sorok];

for(int s=0; s<mtx->sorok; s++) {

mtx->m[s] = new double[mtx->oszlopok];

for(int o=0; o<mtx->oszlopok; o++) {

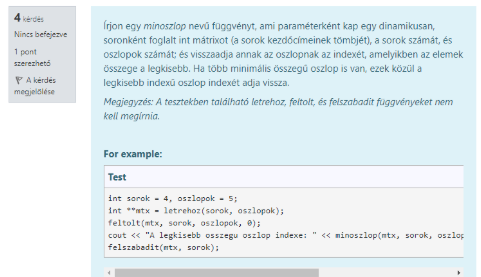
mtx->m[s][o] = m.m[s][o]\*d;

}

}

return mtx;

}



#include <climits>

int minoszlop(int \*\*mtx, int s, int o) {

int minsum = INT\_MAX, mini = -1;

for (int i=0; i<o; i++) {

int sum = 0;

for(int j=0; j<s; j++) {

sum += mtx[j][i];

}

if (sum < minsum) {

minsum = sum;

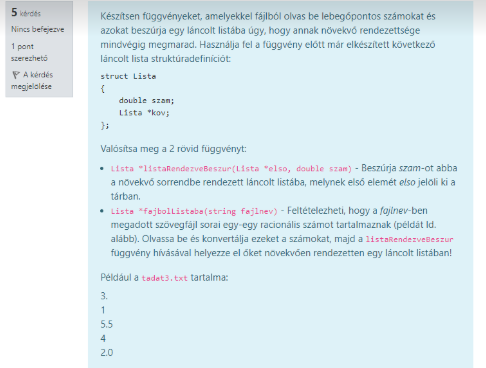
mini = i;

}

}

return mini;

}



struct Lista

{

double szam;

Lista \*kov;

};

// rendezve szúr be

Lista \*listaRendezveBeszur(Lista \*elso, double szam)

{

Lista \*iter, \*uj, \*ideSzur = NULL;

uj = new Lista;

uj->szam = szam;

// az elejére szúrjuk, ha a lista üres, vagy ez a legjobb eredmény

if (elso == NULL || elso->szam > szam)

{

uj->kov = elso;

elso = uj;

}

// nem az elejére beszúrás

else

{

for (iter = elso; iter != NULL && iter->szam <= szam; iter = iter->kov)

ideSzur = iter;

uj->kov = iter;

ideSzur->kov = uj;

}

return elso;

}

// fájlból olvas listába

Lista \*fajbolListaba(string fajlnev)

{

Lista \*ures = NULL;

ifstream fajl(fajlnev.c\_str());

string adat;

if (fajl.is\_open())

{

while (getline(fajl, adat), !fajl.eof())

{

ures = listaRendezveBeszur(ures, atof(adat.c\_str()) );

}

}

return ures;

}