RAČUNALNIŠTVO I
1997/98

Dodatna domača naloga

8. junij 1998

Za dodatno domačo nalogo boste naredili kalkulator za delo z matrikami sestavljenimi iz realnih števil. Operacije, ki jih boste implementirali naj bodo povsem preproste in ni potrebno, da vključujejo postopke za zmanjševanje računskih napak pri računanju s takšnimi matrikami (npr. pivotiranje ipd.). Za lažjo izdelavo domače naloge, si oglejte definicijske, izvedbene in testne module na področju:

`Z:\Predavanja\POLDE` .

Domača naloga je razdeljena na več nalog, ki jih rešujte po vrsti, da boste na koncu dobili celotno rešitev. Prve tri naloge se ukvarjajo z abstraktnimi tipi in jih rešujte v jeziku c. Preostale naloge se ukvarjajo s predmeti in razredi ter jih rešujte v jeziku c++.

1. V prvi nalogi bomo naredili podprograme za branje in pisanje matrike ter nastavljanje vrednosti spremenljivkam abstraktnega tipa `MAT_tMatrika`. Abstraktni tip `MAT_tMatrika` je definiran kot:

   ```
   #define cNajvecjaDimenzija 50   /* najvecja dimenzija matrike */

   typedef struct {
     int   mStolpcev;       /* stolpcev v matriki */
     int   mVrstic;         /* vrstic v matriki */
     float mMatrika[cNajvecjaDimenzija][cNajvecjaDimenzija];
   } MAT_tMatrika;
   ```

   NALOGA: *Napišite podprograma za branje in izpisovanje spremenljivk tipa `MAT_tMatrika`.*

   Podprograma naj bosta del testnega modula, ki bo v zanki bral ukaze s `stdio` dokler ne pride do konca vhodnega toka (`feof`). Možna ukaza sta *Vpiši* (koda `V`) in *Izpiši* (koda `I`). Podprogram za izpis matrike naj izpiše le-to po vrsticah, kjer je vsak element matrike izpisan v formatu `%9.4f`.

   Podprogram za branje naj v naslednjih dveh vrsticah prebere število stolpcev s in število vrstic v v matriki. Po teh dveh vrsticah sledi v krat po s vrstic, ki vsebujejo elemente matrike urejene po vrsticah. Na primer, matriko

   $$\begin{bmatrix} 2.0 & 1.2 & 3.5 \\ -5.7 & 0.6 & 0 \end{bmatrix}$$

   bi zapisali kot

```
3
2
2.0
1.2
3.5
-5.7
0.6
0.0
```

Primer izvajanja testnega programa je naslednji

```
V
3
2
2.0
1.2
3.5
-5.7
0.6
0.0
I
```

Nakar se izpiše

```
   2.0000   1.2000   3.5000
  -5.7000   0.6000   0.0000
```

2. V drugi nalogi bomo naredili izvedbeni modul, ki bo nudil osnovne operacije nad matrikami.

   Naloga: *Napišite izvedbeni modul za naslednji definicijski modul* `MAT`.

   Definicijski modul `MAT` ima obliko:

```
void MAT_Add(const MAT_tMatrika mat1, const MAT_tMatrika mat2,
             MAT_tMatrika * result);
/* Desc:    Sestejemo matriki mat1 in mat2 v result.
   Parms:   mat1 -- prvi sestevanec (po vrednosti)
            mat2 -- drugi sestevanec (po vrednosti)
            result -- vsota (po referenci)
   Result:
   Pre:
   Post:    result= mat1 + mat2
*/

void MAT_Sub(const MAT_tMatrika mat1, const MAT_tMatrika mat2,
             MAT_tMatrika * result);
/* Desc:    Odstejemo od matrike mat1 matriko mat2 in razliko shranimo
            v result.
   Parms:   mat1 -- odstevanec (po vrednosti)
            mat2 -- odstevatelj (po vrednosti)
```

```
            result -- razlika (po referenci)
   Result:
   Pre:
   Post:    result= mat1 - mat2
*/

void MAT_Mult(const MAT_tMatrika mat1, const MAT_tMatrika mat2,
              MAT_tMatrika * result);
/* Desc:   Zmnozimo matriki mat1 in mat2 ter zmnozek shranimo v result.
   Parms:  mat1 -- faktor (po vrednosti)
           mat2 -- faktor (po vrednosti)
           result -- zmnozek (po referenci)
   Result:
   Pre:
   Post:   result= mat1 * mat2
*/
```

Pri posameznih operacijah se lahko zgodi, da se dimenzije matrik ne ujemajo (število stolpcev oziroma vrstic). V tem primeru matriko po potrebi razširite in dopolnite z vrednostmi 0.

3. V tretji nalogi bomo naredili testni modul za modul `MAT` iz druge naloge.

   NALOGA: *Napišite testni modul za modul MAT, ki naj uporablja podobne ukaze kot so jih uporabljali testni moduli v ostalih domačih nalogah.*[1]

   Program naj se obnaša podobno kot preprost namizni kalkulator, ki uporablja le dve števili (matriki): `zadnja` in `predzadnja`. Na začetku imata obe matriki vrednost [0.0]. Nato nad njima lahko opravljamo naslednje operacije (kode ukazov za posamezne operacije so zapisane v oklepajih):

   vpiši (`V`): vrednost predzadnje matrike postane vrednost zadnje matrike (`predzadnja <- zadnja`), novo vrednost zadnje matrike pa preberemo s `stdio` (zaslona);

   seštej (`S`): seštej zadnjo in predzadnjo matriko; potem predzadnji matriki priredimo vrednost zadnje matrike in zadnji vrednost seštevka;

   odštej (`O`): podobno kot seštevanje, le da tokrat od predzadnje matrike odštejemo zadnjo matriko;

   zmnoži (`M`): podobno kot seštevanje, le da tokrat zmnožimo obe matriki;

   zamenjaj (`Z`): medsebojno zamenjaj zadnjo in predzadnjo matriko;

   izpiši (`I`): izpiši vrednost zadnje matrike.

   Operaciji izpisovanja in vpisovanja sta podrobneje definirani že v prvi nalogi in obe naj se obnašata enako kot sta se obnašali tam. Iz prve naloge tudi prevzamite njuni izvedbi.

4. V prejšnjih nalogah smo se ukvarjali z abstraknim tipom `MAT_tMatrika` v jeziku c. V tej nalogi bomo definirali razred matrik `CMat`, ki bo nadomestil abstraktni tip `tMatrika`. Podoben prehod smo naredili v domačih nalogah iz modula `ART` v modul `Cint`.

[1] V resnici lahko za osnovo vzamete testne module iz prejšnjih domačih nalog.

Naloga: *Napišite izvedbeni modul `CMat` in testni modul zanj.*

Razred `CMat` („**C**lass **Mat**rik") vsebuje kot predmete matrike, nad katerimi lahko izvajamo osnovne aritmetične operacije (seštevanja, odštevanja in množenja) ter pretvorbe (prirejanja) iz in v abstraktni tip `MAT_tMatrika`. Razred `CMat` je definiran kot:

```
#include "../../code/MAT/Mat.h"  // abstraktni tip MAT_tMatrika

class CMat {
/* Desc:    Razred matrik.
*/

//-----------------------------------------------------------------------
//---------------------------------------------[ ZASEBNO, NE SKILI!! ]
private:
  MAT_tMatrika mMatrika;    // notranja predstavitev matrike

//-----------------------------------------------------------------------
//-----------------------------------------------------------[ JAVNO ]
public:

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// - - - - - - - - - - - - - - - - - - - - - -[ ustvarjanje / unicevanje ]
  CMat();
/* Desc:   Konstruktor. Ustvari novo matriko z vrednostjo nic.
   Parms:  -
   Result:
   Pre:
   Post:   this = [0.0]
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  CMat(MAT_tMatrika mat);
/* Desc:   Konstruktor. Ustvari novo matriko z vrednostjo mat.
   Parms:  mat -- prepisovana matrika (po vrednosti)
   Result:
   Pre:
   Post:   this = mat
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  void Priredi(CMat mat);
/* Desc:   Matriko, ki jo predstavlja mat, prepisemo v ta predmet.
   Parms:  stev -- prepisovano stevilo (po vrednosti)
   Result:
   Pre:
   Post:   this = mat
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
  ~CMat();
/* Desc:   Destruktor.
   Parms:  -
   Result:
   Pre:
   Post:
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// - - - - - - - - - - - - - - - - - - - - - - - - [ matematicne operacije ]
  void Pristej(CMat mat);
/* Desc:   Pristejemo matriko mat.
   Parms:  mat -- pristevatelj (po vrednosti)
   Result:
   Pre:
   Post:   this'= this + mat
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  void Odstej(CMat mat);
/* Desc:   Odstejemo matriko mat.
   Parms:  mat -- odstevatelj (po vrednosti)
   Result:
   Pre:
   Post:   this'= this - stev
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  void Zmnozi(CMat mat);
/* Desc:   Matriko, ki jo predstavlja ta predmet, zmnozimo s mat.
   Parms:  mat -- faktor (po vrednosti)
   Result:
   Pre:
   Post:   this'= this * stev
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
// - - - - - - - - - - - - - - - - - - - - - - - - - - - - [ poizvedovanja ]
  MAT_tMatrika Vrednost();
/* Desc:   Matriko, ki jo predstavlja ta predmet, vrnemo v
           predstavitvi abstraktnega tipa MAT_tMatrika.
   Parms:  -
   Result: RESULT
   Pre:
   Post:   RESULT = this
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  int Stolpcev();
/* Desc:   Vrnemo stevilo stolpcev v tem predmetu (matriki).
```

```
   Parms:  -
   Result: stevilo stolpcev v this
   Pre:
   Post:
*/

// - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  int Vrstic();
/* Desc:   Vrnemo stevilo vrstic v tem predmetu (matriki).
   Parms:  -
   Result: stevilo vrstic v this
   Pre:
   Post:
*/

}; // CMat
```

NAMIG: Pri izdelavi testnega modula si pomagajte s testnim modulom iz prejšnje naloge in s testnim modulom za `Cint` iz prejšnjih domačih nalog.

5. Na vajah smo napisali program, ki je opravljal funkcije kalkulatorja. Pri tem je uporabljal za svoje delo sklad celih števil. V eni od domačih nalog smo napisali kalkulator za delo z racionalnimi števili, tokrat bomo delali z matrikami.

   NALOGA: *Napišite kalkulator za delo z matrikami.*

   Pri tem uporabite modula `CMat` iz prejšnje domače naloge in `Skladi` (modul z vzorci) z vaj.

   Program naj uporablja enake ukaze kot smo jih definirali na vajah:

   V prebere matriko in jo naloži na sklad (natančna definicija oblike matrike, ki jo beremo, je v prvi nalogi);

   I vzame matriko s sklada in jo izpiše (natančna definicija oblike matrike, ki jo izpišemo, je v prvi nalogi);

   S vzame s sklada dve matriki, ju sešteje in vsoto naloži ponovno na sklad;

   O vzame s sklada prvo matriko, vzame s sklada drugo matriko, odšteje prvo matriko od druge in razliko ponovno naloži na sklad;

   M vzame s sklada dve matriki, ju zmnoži in zmnožek naloži ponovno na sklad;

   NAMIG: Za osnovo vzemite program z vaj in testnimi moduli iz prejšnjih nalog.

6. [Za dodatne točke – ,,bonus naloga"] Razširite razred `CMat`, da bo namesto z realnimi števili (`float`) delal s predmeti iz razreda racionalnih števil poljubne natančnosti `CRac`, katerega definicijskih in izvedbeni modul najdete na področju

   `Z:\Predavanja\POLDE\code\CRac` .

Rok za oddajo: *17. 7. 1998. do 17h* v asistentov predal pri vratarju na FMF ali po elektronski pošti.