**Tema 5**

**Matrici dinamice**

**Deadline**: Vineri, 13 ianuarie 2012, ora 23:55

**Responsabil**

* Mugurel Ionut Andreica, Email: [mugurel.andreica@cs.pub.ro]

**Cunoştinţe necesare**

* lucrul cu fisiere (binare)
* operatii cu vectori
* alocare dinamica a memoriei + pointeri

|  |
| --- |
| **Cuprins**  [[ascunde](javascript:toggleToc())]   * [1 Introducere](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_5#Introducere) * [2 Cerinţa problemei](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_5#Cerin.C5.A3a_problemei) * [3 Exemplu](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_5#Exemplu) * [4 Testarea temei de casa](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_5#Testarea_temei_de_casa) * [5 Evaluare](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_5#Evaluare) |

**Introducere**

O **matrice dinamica** este o matrice avand elemente de tip "int", asupra careia se pot efectua urmatoarele operatii:

* se adauga o linie noua la matrice
* se adauga o coloana noua la matrice
* se extinde matricea la dreapta prin alipirea la ea a unei alte matrici avand acelasi numar de linii (practic, coloanele celei de-a doua matrici se adauga in continuarea coloanelor primei matrici)
* se extinde matricea in jos prin alipirea la ea a unei alte matrici avand acelasi numar de coloane (practic, liniile celei de-a doua matrici se adauga in continuarea liniilor primei matrici)

Fiecare matrice dinamica este stocata pe disc in cate un fisier binar.

**Cerinţa problemei**

Sarcina voastra este sa implementati in cadrul unui fisier header denumit "t5.h" urmatoarele functii:

* **void Init(char\* prefix)**: Aceasta functie realizeaza orice initializari aveti nevoie. Sirul "prefix" indica faptul ca numele oricarui fisier pe care il creati trebuie sa inceapa cu sirul "prefix".
* **int CreateMatrix()**: Aceasta functie creeaza o matrice noua, goala (avand 0 linii si 0 coloane) si ii asociaza un identificator unic (un numar intreg). Identificatorul matricii nou create este intors ca rezultat al functiei.
* **void GetMatrixDimensions(int id, int\* nrows, int\* ncols)**: Aceasta functie seteaza continutul de la adresele de memorie ale pointer-ilor nrows si ncols la numarul de linii, respectiv de coloane, ale matricii avand identificatorul id.
* **void AppendRow(int id, int\* elems)**: Aceasta functie adauga un rand nou la matricea cu identificatorul id. Pointer-ul elems pointeaza catre o zona de memorie ce contine elementele de pe randul nou adaugat (in ordine crescatoare a coloanelor). Numarul de elemente adaugate este egal cu numarul de coloane ale matricii (din momentul adaugarii randului nou).
* **void AppendColumn(int id, int\* elems)**: Aceasta functie adauga o coloana noua la matricea cu identificatorul id. Pointer-ul elems pointeaza catre o zona de memorie ce contine elementele de pe coloana nou adaugata (in ordine crescatoare a liniilor). Numarul de elemente adaugate este egal cu numarul de linii ale matricii (din momentul adaugarii coloanei noi).
* **int\*\* GetSubMatrix1(int id, int start\_row, int start\_col, int end\_row, int end\_col)**: Aceasta functie intoarce o submatrice in memorie ce contine toate elementele dintre liniile start\_row si end\_row (inclusiv) si dintre coloanele start\_col si end\_col (inclusiv) ale matricii dinamice cu identificatorul id. Liniile si coloanele unei matrici dinamice sunt numerotate incepand de la 0.
* **int\*\* GetSubMatrix2(int\*\* base\_submatrix, int start\_row, int start\_col, int end\_row, int end\_col)**: Aceasta functie intoarce o submatrice in memorie ce contine elementele dintre liniile start\_row si end\_row (inclusiv) si dintre coloanele start\_col si end\_col (inclusiv) ale submatricii base\_submatrix. base\_submatrix este o submatrice intoarsa anterior de un apel al functiilor GetSubMatrix1 sau GetSubMatrix2. Liniile si coloanele unei submatrici sunt numerotate incepand de la 0. Submatricea nou creata trebuie sa se suprapuna peste submatricea base\_submatrix. Mai exact, atunci cand se modifica un element al noii submatrici, elementul corespunzator din base\_submatrix trebuie sa se modifice si el in mod automat (si invers).
* **void PutSubMatrix(int\*\* submatrix)**: Scrie elementele submatricii "submatrix" peste elementele corespunzatoare din matricea dinamica ale carei elemente le contine. submatrix a fost intoarsa anterior de un apel al functiilor GetSubMatrix1 sau GetSubMatrix2.
* **void ExtendRight(int id1, int id2)**: Extinde la dreapta matricea dinamica cu identificatorul id1 prin alipirea matricii dinamice cu identificatorul id2. Practic, coloanele matricii dinamice id2 sunt adaugate in continuare coloanelor matricii id1. Cele doua matrici dinamice vor avea acelasi numar de linii in momentul apelului acestei functii.
* **void ExtendDown(int id1, int id2)**: Extinde in jos matricea dinamica cu identificatorul id1 prin alipirea matricii dinamice cu identificatorul id2. Practic, liniile matricii id2 sunt adaugate in continuarea liniilor matricii id1. Cele doua matrici dinamice vor avea acelasi numar de coloane in momentul apelului acestei functii.
* **int GetNumberOfElementsBetweenValues(int id, int vmin, int vmax)**: Intoarce numarul de elemente ale matricii dinamice cu identificatorul id a caror valoare este cuprinsa in intervalul **[vmin,vmax]**.
* **void Finish()**: In cadrul acestei functii veti salva datele necesare in fisiere (sau, in functie de implementare, doar veti inchide fisierele deschise), pentru a fi disponibile la rulari ulterioare. De asemenea, tot in cadrul acestei functii va trebui sa eliberati toate zonele de memorie alocate dinamic (si nedezalocate inca) de functiile voastre.

Pe langa functiile mentionate mai sus, fisierul "t5.h" poate contine orice alte structuri, functii si variabile pe care le veti considera necesare. Pentru declararea unor variabile globale accesibile in functiile din fisierul "t5.h", le veti declara pur si simplu undeva in interiorul fisierului "t5.h" (dar inaintea definirii functiilor ce utilizeaza variabilele respective).

De asemenea, aveti libertatea de a alege cel mai potrivit mod de a reprezenta matricile in cadrul fisierelor binare (cu restrictia ca fiecare matrice sa fie stocata intr-un singur fisier binar, independent).

**Exemplu**

In aceasta sectiune vor fi prezentate, pentru claritate, cateva exemple de apeluri ale functiilor descrise mai sus.

#include <stdio.h>  
#include "t5.h"  
   
int main() {  
 Init("datafile");  
 int mid0 = CreateMatrix();  
 */\* Intrucat matricea are 0 coloane in acest moment, nu conteaza ce parametru dam ca pointer catre zona de memorie ce contine elementele noii linii. \*/*  
 AppendRow(mid0, **NULL**);  
 AppendRow(mid0, **NULL**);  
 int col0[] = { 3, 4 };  
 AppendColumn(mid0, col0);  
 int col1[] = { 5, 6 };  
 AppendColumn(mid0, col1);  
 int mid1 = CreateMatrix();  
 AppendColumn(mid1, **NULL**);  
 int row0[] = { 7 };  
 AppendRow(mid1, row0);  
 int row1[] = { 8 };  
 AppendRow(mid1, row1);  
 int nl0, nc0, nl1, nc1;  
 */\* Trebuie sa obtinem nl0 == 2 si nc0 == 2. \*/*  
 GetMatrixDimensions(mid0, &nl0, &nc0);  
 */\* Trebuie sa obtinem nl1 == 2 si nc1 == 1. \*/*  
 GetMatrixDimensions(mid1, &nl1, &nc1);  
 */\* Intrucat cele 2 matrici au acelasi numar de linii, putem extinde una din ele la dreapta cu cealalta matrice. \*/*  
 */\* Extindem matricea dinamica mid1 la dreapta cu matricea dinamica mid0. In urma extinderii, mid1 va avea 2 linii si 3 coloane. \*/*  
 ExtendRight(mid1, mid0);  
 */\* Extindem in jos matricea dinamica mid1 cu ea insasi. In urma extinderii, mid1 va avea 4 linii si 3 coloane. \*/*  
 ExtendDown(mid1, mid1);  
 */\* Obtinem o submatrice s1 ce contine toata matricea dinamica mid1. \*/*  
 int\*\* s1 = GetSubMatrix1(mid1, 0, 0, 3, 2);  
 */\* Afisam submatricea s1. \*/*  
 int i, j;  
 for (i = 0; i <= 3; i++) {  
 for (j = 0; j <= 2; j++)  
 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("%d ", s1[i][j]);  
 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("**\n**");  
 }  
 */\* Obtinem o submatrice s2 ce se suprapune peste submatricea s1. Mai exact, ea va contine doar liniile 1 si 2 si coloana 1 din s1. \*/*  
 int \*\*s2 = GetSubMatrix2(s1, 1, 1, 2, 1);  
 */\* Daca modificam s2[0][0], va trebui sa se modifice automat si s1[1][1]. \*/*  
 s2[0][0]++;  
 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("%d %d**\n**", s2[0][0], s1[1][1]);  
 */\* Scriem submatricea s2 peste elementele din matricea dinamica peste care se suprapune. \*/*  
 PutSubMatrix(s2);  
 */\* Obtinem o noua submatrice s3, care trebuie sa aiba aceleasi elemente ca si s2. \*/*  
 int \*\*s3 = GetSubMatrix1(mid1, 1, 1, 2, 1);  
 for (i = 0; i <= 1; i++)  
 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("%d**\n**", s2[i][0] - s3[i][0]);  
 */\* Calculam numarul de elemente din mid1 cuprinse intre 4 si 7 (inclusiv). \*/*  
 int num\_elems = GetNumberOfElementsBetweenValues(mid1, 4, 7);  
 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("%d**\n**", num\_elems);  
 Finish();  
 return 0;  
}

**Testarea temei de casa**

Pentru testare se va folosi un program de test, denumit "t5\_test.c". Arhiva ZIP [t5test.zip](http://cs.curs.pub.ro/2011/pluginfile.php/8976/mod_assignment/intro/t5test.zip) contine un director denumit "t5test", cu programul de test, un script "run\_tests.sh" si 3 fisiere utilizate pentru a coordona si verifica comportamentul pe teste ("ans-test1.txt", "ans-test2.txt", "ans-test3.txt"). Pentru lansarea in executie a testelor va trebui sa dezarhivati continutul arhivei, sa copiati fisierul vostru "t5.h" in directorul "t5test" obtinut si apoi sa lansati in executie script-ul "run\_tests.sh". Acest script va executa trei teste care vor verifica corectitudinea si eficienta functiilor implementate de dumneavoastra in header-ul "t5.h".

**ATENTIE!** Versiunea initiala a programului de test (t5\_test.c) continea un mic bug si functiona corect doar daca id-urile asignate matricilor erau consecutive, incepand cu 0. Versiunea corectata se gaseste aici: [t5\_test.c](http://cs.curs.pub.ro/2011/pluginfile.php/8976/mod_assignment/intro/t5_test.c). Utilizati aceasta versiune in locul celei din arhiva zip de test. Daca, insa, tema voastra trece testele cu versiunea initiala a programului de test, atunci va trece testele si utilizand versiunea actualizata.

Testul 2 se bazeaza pe datele scrise la testul 1, iar testul 3 se bazeaza pe datele scrise la testele 1 si 2. De aceea, fisierele binare create/modificate in cadrul unui test trebuie sa poate fi utilizate de functiile dumneavoastra in cadrul testelor urmatoare.

Fiecare test valoreaza 30 puncte (din 100); punctajul pe care il veti primi pentru fiecare test va fi ori 30 (daca rezultatul testului este OK), ori 0 (daca rezultatul testului este EROARE). Restul de 10 puncte se vor acorda pentru README si pentru stilul de programare (lizibilitate, comentarii in cod, etc.). Fiecare test are o limita de timp - daca durata de executie a testului depaseste aceasta limita de timp, atunci rezultatul testului va fi EROARE. Limitele de timp au fost setate la cel putin dublul timpului de executie al unei implementari de referinta care a fost rulata pe calculatorul pe care se va realiza evaluarea temelor (un Intel Core i7, 2.67 GHz).

Datele scrise in timpul celor 3 teste (elementele efective ale matricilor dinamice) au in total aproximativ 2 GB. Implementarea de referinta utilizeaza cu maxim 5% in plus fata de spatiul ocupat de datele efective ale matricilor (pentru reprezentarea matricilor dinamice in fisierele binare). Voi puteti considera ca aveti la dispozitie maxim 3 GB de spatiu pe disc in timpul rularii testelor.

**Evaluare**

* Pe site-ul cursului veti incarca o arhiva care va conține următoarele fișiere:
  + **t5.h**
  + **README** (cu explicații despre soluțiile alese pentru implementarea functiilor)
* Arhiva trebuie sa respecte [regulile de trimitere a temelor](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Regulament_general#Reguli_de_trimitere_a_temelor), cu urmatoarele exceptii:
  + arhiva nu trebuie sa contina un makefile, deoarce nu voi scrieti programul executabil (programul executabil este programul de test), ci doar un header utilizat de programul executabil
  + tema nu trebuie trimisa pe vmchecker (nu va fi evaluata acolo), ci upload-ata pe curs.cs.pub.ro (la sectiunea corespunzatoare temei 5 de la cursul de Programare)