

PROGRAMAREA CALCULATOARELOR

Tema #2 - Simulator de Octave (Alocare dinamică)

Deadline: 07.12.2021, ora 23:55.

Responsabili temă: Alexandru Buzea, Bogdan Rizescu

Echipa temei: Radu Nichita, Remus Neațu, Dorinel Filip, Marius-Răzvan Pricop, Lungu Răzvan, Alexandru Bala

Contents

Actualizări	2
Introducere	2
Enunț	2
Încărcarea în memorie a unei matrice	3
Determinarea dimensiunilor unei matrice	4
Afișarea unei matrice	4
Redimensionarea unei matrice	5
Înmulțirea a două matrice	5
Sortarea matricelor	6
Transpunerea unei matrice	6
Eliberarea memoriei unei matrice	7
Eliberarea tuturor resurselor	7
Alte situații	7
(BONUS) Înmulțirea matricelor folosind algoritmul lui Strassen	8
Exemplu de rulare	9
Regulament	10
Arhivă	10
Checker	11
Punctaj	11
Reguli și precizări	11
Alte precizări	12

Actualizări

- 19.11.2021: A fost adăugată clarificarea referitoare la efectuarea operațiilor aritmetice inclusiv în cazul sumei elementelor (drept criteriu de sortare). Mai exact, inclusiv rezultatul sumei se va calcula modulo $10^4 + 7$.
- 19.11.2021: A fost corectat exemplul pentru operația de redimensionare (indexul matricei căreia i se aplică operația lipsea, iar unii indecși erau invalizi).
- 20.11.2021: Au fost adăugate 2 precizări în regulamentul temei 2, referitoare la lungimea maximă a liniilor din fișierul README (80 de caractere/linie) și la copyright.
- 23.11.2021: A fost adăugată o precizare suplimentară referitoare la împărțirea punctajului în cadrul temei 2 (70p teste de bază + 10p README + 20p coding style + 20p bonusul pentru implementarea algoritmului lui Strassen).
- 25.11.2021: A fost adăugată precizarea conform căreia dacă în urma rezultatului operației de modulo numărul este negativ, se va aduna la acesta numărul cu care se face modulo. Testele au fost actualizate și a fost urcată o nouă variantă a acestora pe platforma Moodle, în arhiva checker-ului.
- 26.11.2021: A fost adăugată precizarea conform căreia după finalizarea tuturor operațiilor trebuie afișat un rând liber. De asemenea, au fost actualizate testele de evaluare.

Introducere

Gigel este student în anul I la ACS. Interacționând cu colegii săi mai mari, acesta află mai multe detalii despre materiile pe care el le va studia semestrul viitor. Printre materii se află și disciplina Metode Numerice, disciplină despre care Gigel a auzit că este foarte importantă și ar vrea să aște mai multe. Printre altele, descoperă și că va lucra într-un environment nou, în care el nu a mai lucrat până acum, denumit **GNU Octave**.

GNU Octave este un mediu de programare (software) ce folosește un limbaj de programare propriu (Octave), limbaj ce este specializat în **lucrul cu matrice**. El reprezintă versiunea open-source a limbajului Matlab (**Matrix Laboratory**). Este folosit pentru implementarea algoritmilor cu specific numeric și dispune de o mare varietate de pachete pe care le puteți folosi (printre altele, pachete de procesarea imaginilor, parsarea fișierelor (I/O), calcul simbolic, procesarea fișierelor audio, biblioteci de algebră liniară etc). Pentru mai multe detalii despre facilitățile oferite, accesați [Octave Forge](#).

Observăm că există diferențe semnificative între GNU Octave și C:

Caracteristici	GNU Octave	C
Tip de limbaj	Interpretat	Compilat
Nivel	Limbaj de nivel înalt	Limbaj de nivel mediu-jos
Alocare de memorie	By default	Explicit

Enunț

Aflând mai multe detalii despre limbajul Octave, Gigel a aflat că limbajul de programare și întreg pachetul software sunt scrise în **C** și **C++**. Fiind o fire foarte inventivă și originală, Gigel decide să își construiască propriul său software numeric pe care îl va folosi la disciplina Metode Numerice, folosind cunoștințele acumulate până acum la disciplina Programarea Calculatoarelor și Limbaje de Programare 1. În primă fază, el va face câteva ipoteze simplificatoare :

- Folosind programul său, scris în limbajul **C**, el va stoca **toate tablourile** folosind **alocarea dinamică**

a memoriei.

- Pentru a evita erorile de aproximare în calculul cu numere reale, precum și eventuale probleme de condiționare ale matricelor (despre care veți învăța la Metode Numerice în semestrul II), matricele stocate vor conține **numai** numere întregi.
- Pentru a evita eventuale probleme de **overflow**, toate valorile numerice obținute ca urmare a operațiilor matematice vor fi calculate modulo $10^4 + 7$. **ATENȚIE!** Dacă rezultatul operației după modulo este un număr negativ, se va aduna la rezultat numărul cu care se face operația de modulo, în cazul nostru $10^4 + 7$.
- După finalizarea tuturor operațiilor trebuie să afișați un rând liber.
- Programul astfel scris va suporta un set de comenzi, care vor fi descrise în ceea ce urmează.

Fiindcă nu are experiență în ceea ce privește crearea unei interfețe grafice, Gigel va folosi linia de comandă (terminalul din Linux) pentru a scrie **comenzi**, de forma **command** **<input>**. Comenzile pe care Gigel își dorește să le implementeze sunt:

- Încărcarea în memorie a unei noi matrice.
- Determinarea dimensiunilor unei matrice.
- Afișarea unei matrice.
- Redimensionarea unei matrice folosind indexarea specifică Matlab/Octave.
- Înmulțirea a 2 matrice.
- Sortarea matricelor în memorie, pe baza sumei tuturor elementelor matricei.
- Transpunerea unei matrice.
- Eliminarea unei matrice din memorie.
- Oprirea programului și dealocarea tuturor resurselor.
- (BONUS) Înmulțirea a două matrice folosind algoritmul lui Strassen.

Încărcarea în memorie a unei matrice

În acest caz, formatul unei comenzi este următorul:

- Se primește obligatoriu caracterul **L** de la STDIN.
- Apoi, se primesc 2 numere naturale, **m** și **n** (cu **m**, **n** ≥ 1), ce reprezintă dimensiunile matricei de citit (**m** reprezintă numărul de linii ale matricei, iar **n** reprezintă numărul de coloane ale matricei)
- În continuare, se citesc de la STDIN **m · n** numere întregi, din intervalul $[-10^4, 10^4]$.

Observații adiționale

- Toate matricele cu care se va lucra vor fi citite în acest mod și stocate în memorie **simultan**; este deci nevoie de crearea unei mulțimi/colecții unde să stocăm matricele noastre.
HINT: Resizable array.
- Atunci când o nouă matrice este citită (o nouă comandă de încărcare este dată), matricea va fi adăugată la finalul listei de matrice.
- Nu există nicio limitare asupra numărului de matrice pe care îl putem încărca în memorie.

Input

```
1 L 4 7
2 3 51 75 3 6 27 2
3 10 20 95 19 21 100 8
4 71 3 32 50 29 40 25
5 12 49 29 40 17 63 29
```

Output

Pentru această comandă, nu se afișează nimic la STDOUT.

Determinarea dimensiunilor unei matrice

În acest caz, formatul unei comenzi este următorul:

- Se primește caracterul **D** de la STDIN.
- Apoi, se primește un singur număr întreg reprezentând **indexul** matricei pentru care se dorește determinarea dimensiunilor.

Observații adiționale

- Se va afișa mesajul de eroare „No matrix with the given index” dacă indexul matricei nu corespunde niciunei matrice încărcată în memorie.

Input

```
1 D 0
```

Output

Se vor afișa dimensiunile matricei (mai întâi numărul de linii, după care numărul de coloane) de la indexul 0 (considerând că avem încărcată deja matricea de la comanda precedentă) sau eventualul mesaj de eroare, dacă nu avem nicio matrice la indexul dat:

```
1 4 7
```

Afișarea unei matrice

În acest caz, formatul unei comenzi este următorul:

- Se primește caracterul **P** de la STDIN.
- Apoi, se primește un singur număr întreg reprezentând **indexul** matricei pentru care se dorește afișarea elementelor matricei.

Observații adiționale

- Se va afișa mesajul de eroare „No matrix with the given index” dacă indexul matricei nu corespunde niciunei matrice încărcată în memorie.

Input

```
1 P 0
```

Output

Se va afișa matricea de la indexul dat (în cazul nostru, indexul 0) sau mesajul de eroare precizat mai sus:

```
1 3 51 75 3 6 27 2
2 10 20 95 19 21 100 8
3 71 3 32 50 29 40 25
4 12 49 29 40 17 63 29
```

Redimensionarea unei matrice

În acest caz, formatul unei comenzi este următorul:

- Se primește caracterul **C** de la STDIN
- Apoi, se primește un singur număr întreg reprezentând **indexul** matricei pentru care se dorește redimensionarea.
- În continuare, se primește un număr întreg care reprezintă **numărul de linii după care se face redimensionarea - l**, iar apoi se primesc **l** indici de linii.
- În continuare, se primește un număr întreg care reprezintă **numărul de coloane după care se face redimensionarea - c**, iar apoi se primesc **c** indici de coloane.

Observații adiționale

- Se va afișa mesajul de eroare „No matrix with the given index” dacă indexul matricei nu corespunde niciunei matrice încărcată în memorie.
- Operația de redimensionare are loc și cu permutarea coloanelor și liniilor, în ordinea dată în input. Pentru exemple suplimentare prin care să înțelegeți cum are loc operația de redimensionare, vizitați [acest document](#).
- Operația de redimensionare va avea loc pe matricea cu indexul dat, **in-place** (modificările vor avea loc la nivel de matrice, direct pe matricea cu indexul dat, fără a mai păstra matricea veche).

Input

```
1 C 0
2 2
3 3 1
4 3
5 6 3 4
```

Output

Nu se va afișa nimic la STDOUT (cu excepția eventualului mesaj de eroare).

Înmulțirea a două matrice

În acest caz, formatul comenzii este următorul:

- Se primește caracterul **M** de la STDIN.
- Apoi, se primesc două numere întregi, reprezentând cei doi indecși ai matricelor pentru care se dorește rezultatul înmulțirii.

Observații adiționale

- Matricea-produs va fi adăugată la finalul mulțimii de matrice.

- Se va afișa mesajul de eroare „No matrix with the given index” dacă indexul vreunei matrice nu corespunde niciunei matrice încărcată în memorie (eroarea se va afișa o singură dată în cazul în care niciunul din indecși nu este valid).
- Se va afișa eroarea „Cannot perform matrix multiplication” dacă înmulțirea matricelor nu poate fi efectuată din cauza dimensiunilor nonconforme ale celor 2 operanzi.

Input

```
1 M 2 3
```

Output

Nu se va afișa nimic la STDOUT, cu excepția eventualelor mesaje de eroare.

Sortarea matricelor

Această comandă își propune să sorteze matricele crescător, în funcție de suma tuturor elementelor sale. Prin **sumă** se înțelege suma tuturor elementelor matricei, modulo $10^4 + 7$. Formatul comenzii este următorul:

- Se primește un singur caracter, și anume caracterul **O** de la STDIN.

Input

```
1 O
```

Output

Nu se va afișa nimic la STDOUT.

Transpunerea unei matrice

În acest caz, formatul comenzii este:

- Se primește caracterul **T** de la STDIN.
- Se primește un număr întreg, reprezentând indexul matricei de transpus.

Observații adiționale

- Se va afișa mesajul de eroare „No matrix with the given index” dacă indexul vreunei matrice nu corespunde niciunei matrice încărcată în memorie (eroarea se va afișa o singură dată în cazul în care niciunul din indecși nu este valid).
- Modificarea va avea loc, de asemenea, **in-place** (modificările vor avea loc la nivel de matrice, direct pe matricea cu indexul dat, fără a mai păstra matricea veche).

Input

```
1 T 0
```

Output

Nu se va afișa nimic la STDOUT, cu excepția eventualelor mesaje de eroare.

Eliberarea memoriei unei matrice

În acest caz, formatul comenzii este:

- Se primește caracterul **F** de la STDIN.
- Se primește un număr întreg, reprezentând indexul matricei a cărei memorie ocupată trebuie eliberată (memoria ocupată de matrice trebuie dealocată, iar matricea trebuie eliminată din mulțime).

Observații adiționale

- Se va afișa mesajul de eroare „No matrix with the given index” dacă indexul dat nu corespunde niciunei matrice încărcată în memorie.
- Indexul corespunzător matricei eliberate nu va rămâne gol (fără matrice), ci toate matricele vor fi deplasate cu o poziție în șirul matricelor (nu se permite existența de goluri).
- Este necesar ca memoria alocată să fie folosită **eficient**. Astfel, la eliberarea prea multor matrice, este recomandat să se realoce memoria alocată dinamic pentru șirul de matrice (exemplu: înjumătățirea acesteia, dacă este posibil).

Input

```
1 F 0
```

Output

Nu se va afișa nimic la STDOUT, cu excepția eventualelor mesaje de eroare.

Eliberarea tuturor resurselor

În acest caz, formatul comenzii este:

- Se primește doar caracterul **Q** de la STDIN.

Observații adiționale

- Primirea comenzii Q reprezintă finalizarea programului (dealocarea tuturor resurselor).

Input

```
1 Q
```

Output

Nu se va afișa nimic la STDOUT.

Alte situații

Dacă caracterul dat nu corespunde niciunei comenzi prezentate mai sus, se va afișa mesajul „Unrecognized command”.

(BONUS) Înmulțirea matricelor folosind algoritmul lui Strassen

În acest caz, formatul comenzii este:

- Se primește caracterul **S** de la STDIN.
- Apoi, se primesc două numere întregi, reprezentând cei doi indcși ai matricelor pentru care se dorește rezultatul înmulțirii.

Observații adiționale

- Matricea-produs va fi adăugată la finalul mulțimii de matrice.
- Pentru simplitate, vom efectua algoritmul lui Strassen numai pe matrice pătratice, ale căror dimensiuni sunt puteri ale lui 2 (fapt garantat de testele primite).
- Se va afișa mesajul de eroare „No matrix with the given index” dacă indexul vreunei matrice nu corespunde niciunei matrice încărcată în memorie (eroarea se va afișa o singură dată în cazul în care niciunul din indcși nu este valid).
- Se va afișa eroarea „Cannot perform matrix multiplication” dacă înmulțirea matricelor nu poate fi efectuată din cauza dimensiunilor nonconforme ale celor 2 operanzi.

Input

1 S 3 4

Output

Nu se va afișa nimic la STDOUT, cu excepția eventualelor mesaje de eroare.

Exemplu de rulare

Fie următorul exemplu de rulare, exemplu care folosește toate comenzile din partea obligatorie a temei noastre:

Input

```
1 L 2 3
2 4330 4243 5453
3 2432 4322 6785
4 L 3 4
5 2312 2657 3292 3587
6 3273 3203 3493 9298
7 2193 2143 6576 6322
8 L 3 3
9 2422 4312 6577
10 6576 5644 4325
11 5435 6456 4576
12 P 2
13 P 1
14 O
15 D 2
16 T 2
17 Z
18 M 2 0
19 P 3
20 C 3
21 3
22 2 0 2
23 2
24 2 0
25 P 3
26 F 3
27 P 3
28 Q
```

Output

```
1 2422 4312 6577
2 6576 5644 4325
3 5435 6456 4576
4 2312 2657 3292 3587
5 3273 3203 3493 9298
6 2193 2143 6576 6322
7 3 4
8 Unrecognized command
9 4560 365 9365
10 8010 9593 5662
11 7111 835 3725
12 8555 3812 152
13 3725 7111
14 9365 4560
15 3725 7111
16 No matrix with the given index
```

Explicație

- Primele comenzi (de la liniile 1, 4, 8) sunt comenzi de tip **load**, așa că matricele sunt doar încărcate, fără a se afișa nimic la STDOUT.
- Cele două comenzi de tip **print** de la liniile 12 și 13 vor afișa cele două matrice corespunzătoare indecșilor introduși.
- Comanda de sortare (**O**) va sorta matricele în memorie în funcție de suma tuturor elementelor din acea matrice (în cazul nostru, va sorta cele 3 matrice încărcate până acum)
- Comanda de la linia 15 va afișa **dimensiunile** matricei cu indexul 2 (care, în cazul nostru, va fi matricea inițial aflată pe indexul 1 - înainte de sortare - întrucât aceasta are cea mai mare sumă a elementelor): 3 și 4 (matrice cu 3 linii și 4 coloane).
- Comanda de la linia 16 va realiza **transpusa** matricei cu indexul 2, deci aceasta va avea acum dimensiunile 4 și 3.
- Comanda **Z** nu există în rândul comenzilor propuse spre implementare, de aceea se va afișa mesajul default „Unrecognized command”.
- Comanda de la linia 18 va realiza **înmulțirea** matricelor cu indecșii 2 și 0 (care sunt compatibile pentru înmulțire, deoarece au dimensiunile 4×3 , respectiv 3×3). Matricea rezultat este stocată la finalul șirului de matrice (la indexul 3).
- Comanda de la linia 19 este de tip **print**. Ea va afișa matricea cu indexul 3 (rezultatul înmulțirii precedente).
- Comanda de la liniile 20-24 este o comandă de **redimensionare**: se primește indexul matricei de redimensionat, iar apoi celelalte date de input (vezi [sintaxa operației de redimensionare](#)).
- Comanda de la linia 25 va **printa** matricea cu indexul 3, tocmai pentru a observa modificările aduse matricei.
- Comanda de la linia 26 va **elibera** matricea redimensionată, iar comanda de la linia 27 printează mesajul așteptat: „No matrix with the given index”.
- Comanda de la linia 28 (**quit**) oprește execuția programului și dealocă toate resursele noastre.

Regulament

Regulamentul general al temelor se găsește pe ocv ([Temele de casă](#)). Vă rugăm să îl citiți integral înainte de continua cu regulile specifice acestei teme.

Arhivă

Soluția temei se va trimite ca o arhivă **zip**. Numele arhivei trebuie să fie de forma

Grupă_NumePrenume_Tema2.zip - exemplu: 311CA_AlexandruBuzea_Tema2.zip.

Arhiva trebuie să conțină în directorul **RĂDĂCINĂ** doar următoarele:

- Codul sursă al programului vostru (fișierele **.c** și eventual **.h**).
- Un fișier **Makefile** care să conțină regulile **build** și **clean**.
 - Regula **build** va compila codul vostru și va genera un singur executabil, denumit **my_octave**
 - Regula **clean** va șterge **toate** fișierele generate la build (executabile, binare intermediare etc).
- Un fișier **README** care să conțină prezentarea implementării alese de voi. **NU** copiați bucăți din enunț.

Arhiva temei **NU** va conține: fișiere binare, fișiere de intrare/ieșire folosite de checker, checkerul, orice alt fișier care nu este cerut mai sus.

Numele și extensiile fișierelor trimise **NU** trebuie să conțină spații sau majuscule, cu excepția fișierului README (care are nume scris cu majuscule și nu are extensie). Nerespectarea oricărei reguli din secțiunea **Arhivă** aduce un punctaj **NUL** pe temă.

Checker

Pentru corectarea acestor teme vom folosi scriptul **check** din arhiva **check_my_octave.zip** din secțiunea de resurse asociată temei. Vă rugăm să citiți **README.md** pentru a ști cum să instalați și utilizați checkerul.

Punctaj

Punctajul maxim ce poate fi obținut este de 100p, dintre care 20p reprezintă punctajul acordat pe coding style, iar restul de 80p reprezintă punctajul pe teste (70p) și punctajul pe README (10p). De asemenea, se pot acorda maxim 20p bonus, pe lângă punctajul de 100p, pentru implementarea **algoritmului lui Strassen**.

ATENȚIE! Ponderea acestei teme reprezintă 1p din nota finală la această materie.

Reguli și precizări

- Punctajul pe teste este cel acordat de **check**, rulat pe **vmchecker**. Echipa de corectare își rezervă dreptul de a depuncta pentru orice încercare de a trece testele fraudulos (de exemplu prin hardcodare).
- Punctajul pe calitatea explicațiilor și a codului se acordă în mai multe etape:
 - **corectare automată**
 - * Checkerul va încerca să detecteze în mod automat probleme legate de coding style și alte aspecte de organizare a codului.
 - * Acesta va puncta cu maxim 20p dacă nu sunt probleme detectate.
 - * Punctajul se va acorda proporțional cu numărul de puncte acumulate pe teste din cele 70p.
 - * Checkerul poate să aplice însă și penalizări (exemplu pentru warninguri la compilare) sau alte probleme descoperite la runtime.
 - * Soluțiile vor fi verificate, la runtime, cu valgrind, astfel încât detectarea de probleme raportate de acest utilitar va anula punctajul pentru testul la rularea căreia se descoperă respectiva problemă.
 - **corectare manuală**
 - * Tema va fi corectată manual și se vor verifica și alte aspecte pe care checkerul nu le poate detecta. Recomandăm să parcurgeți cu atenție tutorialul de **coding-style** de pe ocw.cs.pub.ro.
 - * Codul sursă trebuie să fie însoțit de un fișier README care trebuie să conțină informațiile utile pentru înțelegerea funcționalității, modului de implementare și utilizare a programului. Acesta evaluează, de asemenea, abilitatea voastră de a documenta complet și concis programele pe care le produceți și va fi evaluat, în mod analog coding style-ului, de către echipa de asistenți. În funcție de calitatea documentației, se vor aplica depunctări sau bonusuri.
 - * În fișierul README este necesară respectarea unei lungimi maxime a liniilor (maxim 80 de caractere / linie). De asemenea, fișierul README, dar și toate fișierele sursă vor avea inclus un comentariu de copyright, în care vor fi precizate numele complet, semigrupa și anul universitar. Exemplu: Copyright Buzea Alexandru-Mihai-Iulian 311CAb 2021-2022

- * Deprinderea de a scrie cod sursă de calitate, este un obiectiv important al materiei. Sursele greu de înțeles, modularizate neadecvat sau care prezintă hardcodări care pot afecta semnificativ mentenabilitatea programului cerut, pot fi depunctate adițional.
- * În această etapă se pot aplica depunctări mai mari de 20p.
- * O temă care **NU** compilează cu -Wall -Wextra este depunctată la corectarea manuală cu 5p (punctajul echivalent pentru warnings).

Alte precizări

- Implementarea se va face în limbajul C, iar tema va fi compilată și testată **DOAR** într-un mediu **LINUX**. Nerespectarea acestor reguli aduce un punctaj **NUL**.
- Tema trebuie trimisă sub forma unei arhive pe site-ul cursului curs.upb.ro și pe vmchecker.
- Tema poate fi submită de oricâte ori fără depunctări până la deadline-ul hard, existând o penalizare de 10% din punctajul temei pentru fiecare zi dintre deadline-ul soft și momentul trimiterii temei. După deadline-ul hard, **tema nu mai poate fi submită**. Mai multe detalii se găsesc în regulamentul de pe ocw.
- O temă care **NU** compilează pe **vmchecker** **NU** va fi punctată.
- O temă care compilează, dar care **NU** trece niciun test pe **vmchecker**, **NU** va fi punctată.
- Punctajul pe teste este cel acordat de **check** rulat pe **vmchecker**. Echipa de corectare își rezervă dreptul de a depuncta pentru orice încercare de a trece testele fraudulos (de exemplu prin hardcodare).
- Ultima temă submită pe vmchecker poate fi rulată de către responsabili de mai multe ori în vederea verificării faptului că nu aveți buguri în sursă. Vă recomandăm să verificați **local** tema de mai multe ori pentru a verifica că punctajul este mereu același, apoi să încărcați tema.