

# PROGRAMAREA CALCULATOARELOR

## Tema #2 Alocare dinamică

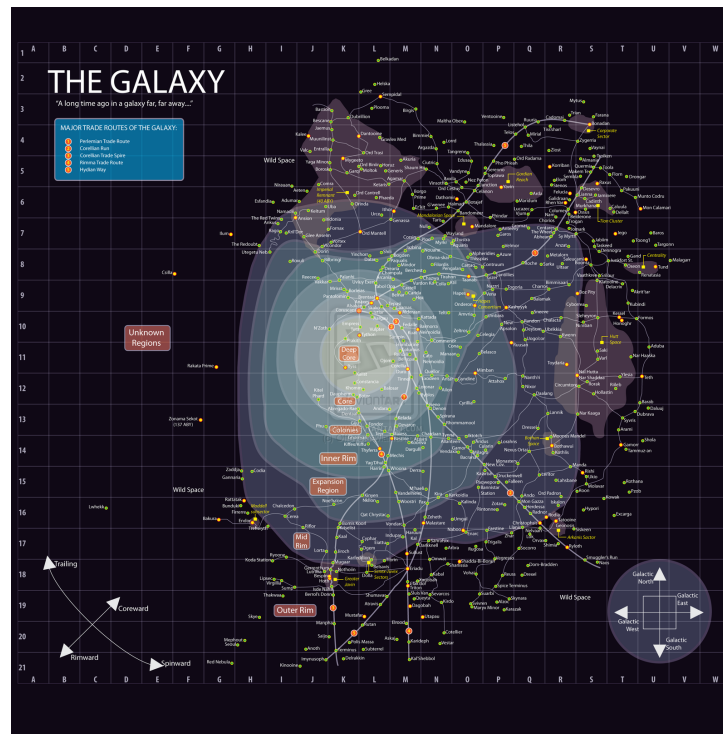
Termen de predare: 01-12-2020 23:55

Responsabil Temă: Ștefania Ghiță

### Contents

Introducere	2
Task 1	3
Task 2	3
Task 3	5
Regulament	8
Arhivă . . . . .	8
Checker . . . . .	8
Punctaj . . . . .	8
Reguli și precizări . . . . .	8
Alte precizări . . . . .	9

## Introducere



Este perioada unui război civil.  
 Navele rebele, atacând dintr-o bază secretă  
 au câștigat prima lor victorie  
 împotriva maleficului Imperiu Galactic.  
 În timpul atacului, spionii rebeli  
 au reușit să fure planurile secrete  
 ale unei arme supreme a Imperiului:  
 Steaua Morții, o stație spațială blindată  
 dotată cu destulă putere pentru a  
 distruge o întreagă planetă.  
 Urmărită de sinistrii agenți Imperiali,  
 Prințesa Leia fuge spre casă  
 la bordul navei sale  
 deținând planurile furate  
 care ar putea salva poporul său  
 și reinstaura libertatea în galaxie...

Odata ce au obtinut planurile, cavalerii Jedi din Imperiul Automaticii si Calculatoarelor trebuie sa le descifreze. Se pare ca Steaua Mortii stie intreg planul galaxiei, iar Leia a reusit sa obtina o harta codificata pe care a salvat-o bunul ei prieten robot R2-D2. Nemaifiind in tineretile sale, androidul a pierdut niste informatii importante.

Pentru a evada, cei mai iscusiti programatori, studentii de la Automatica si Calculatoare, trebuie sa reconstruiasca harta galactica si sa-i ghideze pe rebeli printre obstacolele ce ii asteapta. Pentru a reusi, vor trebui sa studieze cu atentie harta primita, reprezentata din mai multe insiruiiri de numere intregi.

## Task 1

### Enunț

Primul număr primit,  $n$ , reprezintă numărul de linii din imaginea care trebuie reconstruită. Pe următoarele  $n$  linii se regasesc informații despre pixelii imaginii. Astfel, pe o linie  $i$ , se afla:  $m_i, a_{i0}, a_{i1}, a_{i2}, a_{i3}, \dots, a_{im-1}$  unde  $m_i$  e numărul de elemente de pe o linie și  $a_{ij}$  sunt elementele, numere de tipul  $0 \leq i < n, 0 \leq j < m$

Primul lucru care trebuie aflat este dacă scutul protector care înconjoară galaxia este suficient de încărcat. Acesta este definit de bytes-ii de pe conturul hărții. Gradul de încărcare este definit de media aritmetică a acestora. Se dorește afișarea la stdout a acestei medii. Pentru ca se urmărește o precizie cât mai bună pentru rezolvarea acestei misiuni, este impus ca media aritmetică să fie calculată cu 8 zecimale exacte.

Atentie!

- În rezolvarea acestei teme toți bytes-ii se consideră ca fiind numere cu semn. Mai multe detalii, puteți găsi [aici](#)

## Task 2

### Enunț

Urmează corectarea imaginii pentru a obține harta inițială. În continuarea hărții de la Task-ul 1, se regasesc o valoare,  $k$ , reprezentând numărul de modificări, urmată de o însiruire de  $k$  instrucțiuni ce trebuie aplicate pentru a face modificările necesare.

O modificare este codificată astfel: Operatie DimensiuneDate Linie IndexDate [ValoareNoua]

Operatie = Corectarea care trebuie aplicată, reprezentată printr-o literă ce poate fi una din următoarele opțiuni:

- "M" - modify
- "D" - delete
- "S" - swap

DimensiuneDate = Dimensiunea porțiunii de date ce trebuie corectate. Poate fi de tipul:

- "C" - char (1 byte)
- "S" - short (2 bytes)
- "I" - int (4 bytes)

Linie = Linia unde se vor corecta date.

IndexDate = Indexul porțiunii de date ce trebuie corectate. IndexDate este minim 0 pentru operația Swap și minim 1 pentru Modify și Delete. Observați exemplele de mai jos.

ValoareNoua = Valoarea ce va apărea în locul datelor vechi pentru a face corectia. Aceasta valoare este dată doar în cazul operațiilor de modificare, dimensiunea valorii corespunzând cu tipul de date menționat prin DimensiuneDate.

Pentru a înțelege mai bine, iată câteva exemple:

1	M	S	3	5	4563
---	---	---	---	---	------

Se va modifica elementul de pe linia a treia a hărții, aflat la poziția celui de-al cincilea bloc de date de tipul short int. Numărul din acest bloc se va înlocui cu numărul 4563.

Atentie!

- Pentru operatia de modificare, trebuie tratat cu atentie cazul in care zona se afla in afara hartii existente. In acest caz, se vor umple cu zerouri spatiile ramase intre vechea dimensiune si noii octeti modificati, avand grija ca formatul hartii sa fie compatibil cu dimensiunea tipului de date int. Sa presupunem, pentru exemplul nostru, ca linia a treia a hartii continea un singur element de tipul int inainte de modificare. In acest caz, va fi introdus un int cu valoarea 0 pe pozitia 1 a liniei, apoi un al doilea int, care va avea primii doi octeti corespunzatori short-ului introdus, iar urmatorii doi zero.
- Cu alte cuvinte, daca  $\text{index} * \text{dimensiune}(\text{in bytes}) > \text{dimensiunea liniei}(\text{in bytes})$ , toti bytes-ii, incepand cu dimensiunea liniei + 1 si terminand cu  $\text{index} * \text{dimensiune}(\text{in bytes}) - 1$ , vor contine valoarea zero.
- Pentru aceasta operatie, IndexDate incepe de la 1.

1 D C 0 5

Se va sterge de pe linia zero a hartii cel de-al cincilea bloc de date cu dimensiunea char.

Atentie!

- Se garanteaza ca aceasta operatie se poate executa mereu in testele date.
- Prin operatia de stergere, se intelege zeroizarea acelei zone din harta.
- Pentru aceasta operatie, IndexDate incepe de la 1.

1 S I 2 4

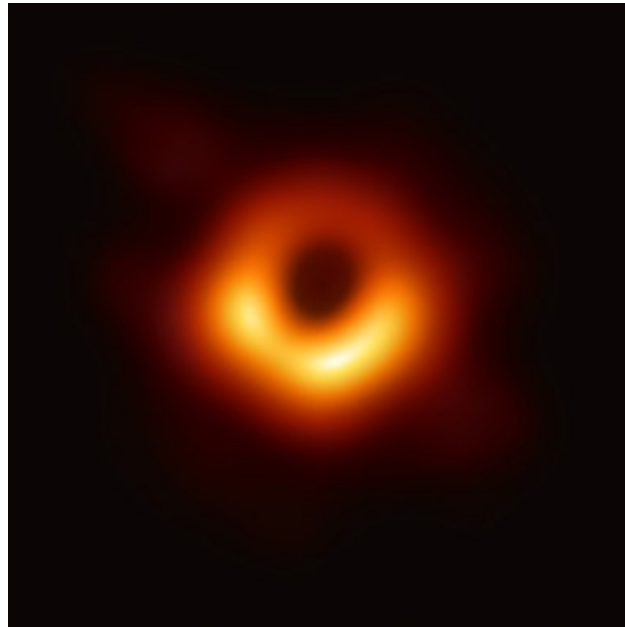
Se va inversa ordinea bytes-ilor din cel de-al patrulea bloc de date cu dimensiunea int de pe linia a doua a hartii.

Atentie!

- Se garanteaza ca aceasta operatie se poate executa mereu in testele date.
- Pentru aceasta operatie, IndexDate incepe de la 0.

Pentru acest task, se cere afisarea hartii dupa aplicarea tuturor celor k modificari.

## Task 3



### Enunț

Acum ca harta a fost reconstruita, mai este nevoie doar de un plan de evadare. Se știe că zonele ce conțin zero reprezintă găuri negre. Pentru a reuși în această misiune, rebelii trebuie să știe locația celei mai mari găuri negre. O gaură neagră supermasivă reprezintă mai multe găuri negre cu proprietatea că din oricare se poate ajunge la oricare trecând numai prin alte găuri negre. Trecerea dintr-o gaură neagră în alta se poate face în oricare din direcțiile sus, jos, stânga sau dreapta (nu și pe diagonală). Supermasivitatea este reprezentată de numărul de găuri negre incluse într-o reuniune.

Se cere locația celei mai mari găuri negre și supermasivitatea acesteia. Prin locație se înțelege gaură neagră componentă cu  $i$  minim,  $j$  minim, în această ordine.

Atenție!

- În cazul în care există mai multe găuri negre cu aceeași supermasivitate, se vor afișa datele pentru cea care are locația la  $i$  minim,  $j$  minim.
- Harta în care se va căuta gaură este cea obținută după rezolvarea task-ului 2.

Harta trebuie stocată astfel încât la orice moment fiecare linie să aibă exact numărul de bytes minim necesari în acel moment, alocându-se/relocându-se la nevoie fiecare linie. Încălcarea acestei reguli sau stocarea statică a hărții duce la un punctaj nul pe tema.

**Date de intrare**

De la tastatură veți primi pe prima linie variabila  $n$ , reprezentând numărul de linii ale imaginii. Pe următoarele  $n$  linii veți primi variabila  $m$ , reprezentând numărul de coloane de pe linia  $i$ , și un șir de  $m$  numere de tipul `int` în format hexazecimal. După aceste  $n$  linii, veți primi variabila  $k$ , reprezentând numărul de comenzi. Pe următoarele  $k$  linii se regăsesc comenzile în formatul de la Task-ul 2.

**Date de ieșire**

Pe prima linie se va afișa mesajul "task 1". Veți afișa pe următoarea linie gradul de încărcare al scutului protector, cerut la TASK-ul 1. Pe cea de-a treia linie se va afișa mesajul "task 2", iar pe următoare  $n$  linii veți afișa configurația hărții după ce ați aplicat modificările de la TASK-ul 2. Pe următoarea linie veți afișa mesajul "task 3", iar pe ultima linie se vor afișa variabilele  $i$ ,  $j$ ,  $s$ , reprezentând linia și coloana locației gaurii negre supermasive, urmate de supermasivitatea acesteia.

Toate datele se vor afișa la **STDOUT**.

**Exemplu****Date de intrare**

```

1 4
2 4 00000001 00000002 00000003 00000004
3 3 00000100 00000101 00000102
4 1 FFFFFFFF
5 2 000000FF 00000100
6 5
7 M S 2 3 00000101
8 M C 3 13 000000FF
9 M I 3 2 00000001
10 D C 2 3
11 S I 1 1

```

**Date de ieșire**

```

1 task 1
2 0.2857143
3 task 2
4 00000001 00000002 00000003 00000004
5 00000100 01010000 00000102
6 FF00FFFF 00000101
7 000000FF 00000001 00000000 000000FF
8 task 3
9 0 1 14

```

**Explicație**

```

1 Initial:
2 0x01 0x00 0x00 0x00 | 0x02 0x00 0x00 0x00 | 0x03 0x00 0x00 0x00 | 0x04 0x00 0x00 0x00
3 0x00 0x01 0x00 0x00 | 0x01 0x01 0x00 0x00 | 0x02 0x01 0x00 0x00
4 0xFF 0xFF 0xFF 0xFF
5 0xFF 0x00 0x00 0x00 | 0x00 0x01 0x00 0x00
6
7 0x01 + 0x02 + 0x03 + 0x04 + 0xFF + 0x01 + 0xFF + 0xFF = 1 + 2 + 3 + 4 - 1 + 1 - 1 - 1
8   = 8
9 nr de bytes pe margine = 28
10 8/28 = 0.2857143
11

```

```

12 Dupa fiecare modificare:
13 M S 2 3 0x101
14
15 0x01 0x00 0x00 0x00 | 0x02 0x00 0x00 0x00 | 0x03 0x00 0x00 0x00 | 0x04 0x00 0x00 0x00
16 0x00 0x01 0x00 0x00 | 0x01 0x01 0x00 0x00 | 0x02 0x01 0x00 0x00
17 0xFF 0xFF 0xFF 0xFF | 0x01 0x01 0x00 0x00
18 0xFF 0x00 0x00 0x00 | 0x00 0x01 0x00 0x00
19
20
21 =====
22 M C 3 13 0xFF
23
24 0x01 0x00 0x00 0x00 | 0x02 0x00 0x00 0x00 | 0x03 0x00 0x00 0x00 | 0x04 0x00 0x00 0x00
25 0x00 0x01 0x00 0x00 | 0x01 0x01 0x00 0x00 | 0x02 0x01 0x00 0x00
26 0xFF 0xFF 0xFF 0xFF | 0x01 0x01 0x00 0x00
27 0xFF 0x00 0x00 0x00 | 0x00 0x01 0x00 0x00 | 0x00 0x00 0x00 0x00 | 0xFF 0x00 0x00 0
   x00
28 =====
29 M I 3 2 0x01
30
31 0x01 0x00 0x00 0x00 | 0x02 0x00 0x00 0x00 | 0x03 0x00 0x00 0x00 | 0x04 0x00 0x00 0x00
32 0x00 0x01 0x00 0x00 | 0x01 0x01 0x00 0x00 | 0x02 0x01 0x00 0x00
33 0xFF 0xFF 0xFF 0xFF | 0x01 0x01 0x00 0x00
34 0xFF 0x00 0x00 0x00 | 0x01 0x00 0x00 0x00 | 0x00 0x00 0x00 0x00 | 0xFF 0x00 0x00 0x00
35 =====
36 D C 2 3
37
38 0x01 0x00 0x00 0x00 | 0x02 0x00 0x00 0x00 | 0x03 0x00 0x00 0x00 | 0x04 0x00 0x00 0x00
39 0x00 0x01 0x00 0x00 | 0x01 0x01 0x00 0x00 | 0x02 0x01 0x00 0x00
40 0xFF 0xFF 0x00 0xFF | 0x01 0x01 0x00 0x00
41 0xFF 0x00 0x00 0x00 | 0x01 0x00 0x00 0x00 | 0x00 0x00 0x00 0x00 | 0xFF 0x00 0x00 0x00
42 =====
43 S I 1 1
44
45 0x01 0x00 0x00 0x00 | 0x02 0x00 0x00 0x00 | 0x03 0x00 0x00 0x00 | 0x04 0x00 0x00 0x00
46 0x00 0x01 0x00 0x00 | 0x00 0x00 0x01 0x01 | 0x02 0x01 0x00 0x00
47 0xFF 0xFF 0x00 0xFF | 0x01 0x01 0x00 0x00
48 0xFF 0x00 0x00 0x00 | 0x01 0x00 0x00 0x00 | 0x00 0x00 0x00 0x00 | 0xFF 0x00 0x00 0x00
49
50 Matrice dupa modificari:
51 0x01 0x00 0x00 0x00 | 0x02 0x00 0x00 0x00 | 0x03 0x00 0x00 0x00 | 0x04 0x00 0x00 0x00
52 0x00 0x01 0x00 0x00 | 0x00 0x00 0x01 0x01 | 0x02 0x01 0x00 0x00
53 0xFF 0xFF 0x00 0xFF | 0x01 0x01 0x00 0x00
54 0xFF 0x00 0x00 0x00 | 0x01 0x00 0x00 0x00 | 0x00 0x00 0x00 0x00 | 0xFF 0x00 0x00 0x00
55
56 Cea mai mare gaura neagra se afla la pozitia [0,1] si are dimensiunea 14, fiind
   formata din bytes-ii de pe pozitiile: [0,1], [0,2], [0,3], [1, 2], [1, 3], [1, 4],
   [1, 5], [0, 5], [0, 6], [0, 7], [2, 2], [4, 1], [4, 2], [4, 3]

```

## Regulament

Regulamentul general al temelor se găsește pe ocw (**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: 317CA\_StefaniaGhita\_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 executabilul **star\_dust**
  - 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 este are nume scris cu majuscule și nu are extensie).

Nerespectarea oricărei reguli din secțiunea **Arhivă** aduce un punctaj **0** pe temă.

## Checker

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

## Punctaj

Distribuirea punctajului:

- Task 1: 22.5p
- Task 2: 30p
- Task 3: 22.5p
- Claritatea și calitatea codului: 20p
- Claritatea explicațiilor din **README**: 5p

ATENȚIE! Punctajul maxim obținut pe temă este de 100p. Acesta 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 80p.
- \* 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 prinde. Recomandăm să parcurgeți cu atenție tutorialul de coding style de pe ocw ([Coding Style](#)).
  - \* 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 CS, de către echipa de asistenți. În funcție de calitatea documentației, se vor aplica depunctări sau bonusuri.
  - \* 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

- Tema trebuie trimisă sub forma unei arhive pe site-ul cursului curs.upb.ro și pe vmchecker.
- Tema poate fi submitată de oricâte ori fără depunctări până la deadline. 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 submitată 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.