

# The Galactic War - SD version

---

## Responsabili:

- Radu Nichita [mailto:radunichita99@gmail.com]
- Remus Neațu [mailto:neaturemus07@gmail.com]
- Miruna-Elena Banu [mailto:mirunaelena.banu@gmail.com]
- Data publicării: **22.03.2021**
- Deadline **HARD: 11.04.2021 23:55:00**

## Actualizări

---

- **03.04.2021** - precizare număr maxim caractere pentru nume planetă

## Obiective

---

- Aprofundarea cunoștințelor în utilizarea limbajului C.
- Implementarea și utilizarea structurii de date **vector circular** [[https://en.wikipedia.org/wiki/Circular\\_buffer](https://en.wikipedia.org/wiki/Circular_buffer)].
- Familiarizarea cu implementarea unei **structuri de date generice**.

## Introducere

---

"Look here, Hardin, you were on Anacreon once before. You were young then; we were both young. But even then we had entirely different ways of looking at things. You're what they call a man of peace, aren't you?

I suppose I am. At least, I consider violence an uneconomical way of attaining an end. There are always better substitutes, though they may sometimes be a little less direct.

Yes. I've heard of your famous remark: "Violence is the last refuge of the incompetent." And yet... I wouldn't call myself exactly incompetent. [...]

Your highness, you are wasting time. If you mean to declare war, and are informing me of the fact, you will allow me to communicate with my government at once.

Sit down, Hardin. I am not declaring war, and you are not communicating with your government at all. When the war is fought, not declared, Hardin, **fought**, the Foundation will be informed of it in due time by the atom blasts of the Anacreonian navy under the lead of my own son upon the flagship *Wienis*."

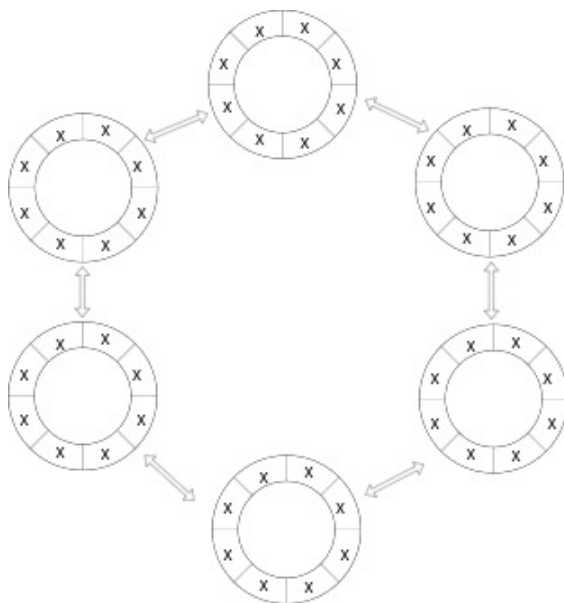
## Cerință

---

Veți avea de simulat un război galactic, în care participanții sunt planetele, fiecare având o linie de scuturi pentru protecție. Planetele sunt așezate într-un cerc, astfel încât fiecare planetă are 2 vecini (în cazul în care există cel puțin 3 planete în galaxie).

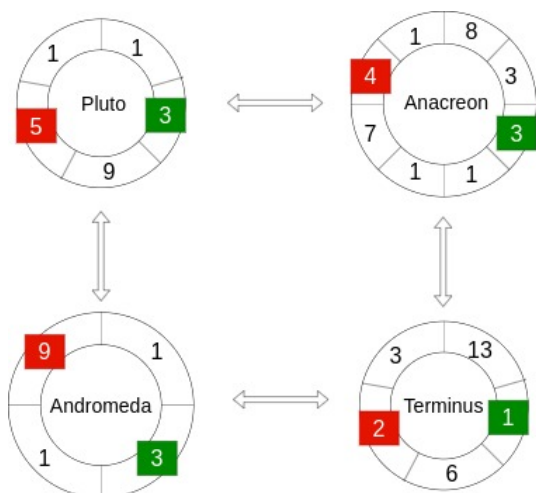
Galaxia va trebui reprezentată cu ajutorul unei Liste Circulare Dublu Înălțuită (CDLL). De asemenea, scuturile fiecărei planete vor fi reprezentate tot cu ajutorul unei CDLL. Prin urmare, **este nevoie să implementați structura CDLL generică** (să poată stoca diferite tipuri de valori).

O reprezentare grafică este următoarea:



- Fiecare planeta are următoarele caracteristici:
  - nume.

- vecin stânga (prev)
- vecin dreapta (next).
- scuturi (numărul scuturilor diferă de la o planetă la alta).
- numărul de planete distruse.



În reprezentarea de mai sus, se afla 4 planete:

- Planeta 1:
  - Nume: Pluto
  - Vecin stanga: Andromeda
  - Vecin dreapta: Anacreon
  - Scuturi: 1 3 9 5 1
- Planeta 2:
  - Nume: Anacreon
  - Vecin stanga: Pluto
  - Vecin dreapta: Terminus
  - Scuturi: 8 3 3 1 1 7 4 1
- Planeta 3:
  - Nume: Terminus
  - Vecin stanga: Anacreon
  - Vecin dreapta: Andromeda
  - Scuturi: 13 1 6 2 3
- Planeta 4:
  - Nume: Andromeda
  - Vecin stanga: Terminus
  - Vecin dreapta: Pluto
  - Scuturi: 1 3 1 9

Inițial, toate planetele au 0 kill-uri (Nicio planetă nu a distrus nicio planetă).

În timpul războiului, o planetă se poate lupta (ciocni) doar cu vecinii săi, astfel, în urma luptei, unele dintre planete vor fi distruse. Atunci când 2 planete să ciocnesc, fiecare își pune la bătaie o unitate de scut. Când o planetă (**p1**) cauzează moartea altei planete (**p2**), numărul de planete distruse (kill-uri) al planetei **p1** crește cu o unitate.

Planetele vor avea punctele de coliziune la indecșii  $n / 4$  și la  $(n / 4) * 3$ , unde  $n$  este numărul de scuturi al planetei. De exemplu, dacă o planetă are 12 scuturi, numerotate de la 0 la 11, atunci vom avea punctul de coliziune cu planeta din dreapta (next) la scutul cu indexul 3 ( $12 / 4$ ) și punctul de coliziune cu planeta din stânga (prev) la scutul cu indexul 9 ( $12 / 4 * 3$ ). În reprezentarea grafică de mai sus, punctul de coliziune al planetei cu vecinul drept are culoarea verde, iar cel cu vecinul stâng are culoarea roșie.

## Input

Pe prima linie se găsește **M**, reprezentând numărul de acțiuni ce vor fi realizate în galaxie. Pe următoarele **M** linii se găsesc informații despre fiecare acțiune. Acțiunile pot fi următoarele:

- **ADD <nume\_planetă> <indice\_planetă> <număr\_scuturi>**
  - se adaugă în galaxie planeta <nume\_planetă> pe poziția <indice\_planetă> (initial galaxia este goală).
  - numele unei planete poate avea maxim **20 de caractere**.
  - această planetă va avea <număr\_scuturi> scuturi (initial toate scuturile au valoarea 1).
  - se va afișa mesajul **The planet <planet\_name> has joined the galaxy.**
  - dacă <indice\_planetă> este mai mare decât numărul de planete din galaxie, se va afișa mesajul: **Planet out of bounds!**
- **BLH <indice\_planetă>**
  - planeta aflată la poziția <indice\_planetă> este înghițită de o gaură neagră și este scoasă din galaxie.
  - se va afișa: **The planet <planet\_name> has been eaten by the vortex.**, unde <planet\_name> este numele planetei aflate la poziția <indice\_planetă> în galaxie.

- dacă <indice\_planeta> este mai mare decât numărul de planete din galaxie, se va afișa mesajul: **Planet out of bounds!**
- **UPG <indice\_planeta> <indice\_scut> <valoare\_upgrade>**
  - planeta aflata la pozitia <indice\_planeta> va adăuga <valoare\_upgrade> la scutul de pe poziția <indice\_scut>.
  - se garantează că <valoare\_upgrade> este un număr pozitiv.
  - dacă <indice\_planeta> este mai mare decât numărul de planete din galaxie, se va afișa mesajul: **Planet out of bounds!**
  - dacă <indice\_scut> este mai mare decât numărul de scuturi ale planetei, se va afișa mesajul: **Shield out of bounds!**
- **EXP <indice\_planeta> <valoare\_scut>**
  - planeta aflata la pozitia <indice\_planet> va adăuga un scut cu valoarea <valoare\_scut> la finalul listei de scuturi.
  - dacă <indice\_planeta> este mai mare decât numărul de planete din galaxie, se va afișa mesajul: **Planet out of bounds!**
- **RMV <indice\_planeta> <indice\_scut>**
  - planeta aflata la pozitia <indice\_planeta> isi va pierde scutul de pe pozitia <indice\_scut>.
  - de ce ar vrea o planeta sa faca asta? pentru că își poate modifica punctele de contact planetele vecine.
  - dacă <indice\_planeta> este mai mare decât numărul de planete din galaxie, se va afișa mesajul: **Planet out of bounds!**
  - dacă <indice\_scut> este mai mare decât numărul de scuturi ale planetei, se va afișa mesajul: **Shield out of bounds!**
  - o planetă nu poate avea mai puțin de 4 scuturi. În cazul în care o planetă are fix 4 scuturi și se încearcă ștergerea unui scut, se va afișa mesajul: **A planet cannot have less than 4 shields!**
- **COL <indice\_planeta1> <indice\_planeta2>**
  - se realizează o coliziune între planetele aflate la pozitiile <indice\_planeta1> (planet1) și <indice\_planeta2> (planet2).
  - se garantează mereu ca planet1 este vecină cu planet2 și este înaintea acesteia în galaxie
    - planet1→next = planet2
  - planetele se vor ciocni în 1 unitate de scut. Scutul lui planet1 ce se va ciocni va fi elementul de pe poziția size / 4 din lista de scuturi, iar scutul lui planet2 ce se va ciocni va fi elementul de pe poziția (size / 4) \* 3, unde size reprezintă dimensiunea listei de scuturi a planetei (indexare de la 0).
  - la coliziune, fiecare scut afectat va pierde o unitate.
  - când o planetă face coliziune într-un loc în care scutul are valoarea 0, planeta face implozie și este eliminată din galaxie. Se va afișa mesajul: **The planet <planet\_name> has imploded.**
  - dacă ambele planete au scuturi în punctul de ciocnire cu valoarea 0, vor imploda amândouă, prima oară afișându-se mesajul pentru planet1.
  - în cazul în care o planeta moare, se va crește numărul de kill-uri (planete distruse) al planetei ramase în viață cu 1 unitate
  - dacă <indice\_planeta1> sau <indice\_planeta2> sunt mai mari decât numărul de planete din galaxie, se va afișa mesajul: **Planet out of bounds!**
- **ROT <indice\_planeta> <sens> <unitati>**
  - se rotește planeta aflata la pozitia <indice\_planeta> cu un număr de <unitati> unitati în direcția <sens>.
  - <sens> este un caracter ce poate avea valorile:
    - c (caz în care se va roti în sensul acelor de ceasornic)
    - t (caz în care se va roti în sens trigonometric).
  - dacă <indice\_planeta> este mai mare decât numărul de planete din galaxie, se va afișa mesajul: **Planet out of bounds!**
  - orice alt caracter în afară de c și t va fi considerat invalid, iar în acest caz se va afișa: **Not a valid direction!**
  - rotirea trebuie realizată eficient. De exemplu, dacă o planetă cu 10 scuturi trebuie să se rotească cu 1043 de unitați, se obține același rezultat ca în cazul în care se realizează rotirea cu 3 unitați.
- **SHW <indice\_planeta>**
  - afișează câteva informații despre planeta aflata la poziția <indice\_planetă> în galaxie
    - numele planetei
    - planetele vecine
    - lista de scuturi
    - câte planete a "ucis"
  - exemplu
    - **NAME:** <nume\_planeta>
    - **CLOSEST:** <nume\_vecin\_stanga> (prev) and <nume\_vecin\_dreapta> (next)
    - **SHIELDS:** <valoare\_shield0> <valoare\_shield1> <valoare\_shield2> .....
    - **KILLED:** <numarul\_de\_planete\_distruse>
  - dacă o planetă are un singur vecin (sunt doar 2 planete în galaxie), se va afișa:
    - **CLOSEST:** <nume\_vecin>
  - dacă o planeta nu are vecini (există o singura planeta în galaxie), se va afișa:
    - **CLOSEST:** none

## Exemplu

### Input

```
22
SHW 0
ADD Mercury 0 4
ADD Venus 1 8
ADD Earth 2 12
ADD Jupiter 3 4
ADD Saturn 4 12
ADD Uranus 5 8
ADD Neptunus 6 5
ADD Pluto 7 9
```

```
BLH 6
ROT 2 c 5
ROT 3 t 4
SHW 2
UPG 4 2 1
COL 3 4
SHW 3
ROT 4 c 1
COL 3 4
SHW 3
SHW 6
ADD Tiaria 4 4
SHW 4
```

## Output

```
Planet out of bounds!
The planet Mercury has joined the galaxy.
The planet Venus has joined the galaxy.
The planet Earth has joined the galaxy.
The planet Jupiter has joined the galaxy.
The planet Saturn has joined the galaxy.
The planet Uranus has joined the galaxy.
The planet Neptunus has joined the galaxy.
The planet Pluto has joined the galaxy.
The planet Neptunus has been eaten by the vortex.
NAME: Earth
CLOSEST: Venus and Jupiter
SHIELDS: 1 1 1 1 1 1 1 1 1 1 1
KILLED: 0
NAME: Jupiter
CLOSEST: Earth and Saturn
SHIELDS: 1 0 1 1
KILLED: 0
The planet Jupiter has imploded.
NAME: Saturn
CLOSEST: Earth and Uranus
SHIELDS: 1 1 1 2 1 1 1 1 1 0 0 1
KILLED: 1
Planet out of bounds!
The planet Tiaria has joined the galaxy.
NAME: Tiaria
CLOSEST: Saturn and Uranus
SHIELDS: 1 1 1 1
KILLED: 0
```

## Explicații

Avem 22 de comenzi. inițial, galaxia este goală.  
Comanda de show eșuază, deoarece nu avem nicio planetă în galaxie.  
Se adaugă planeta Mercury pe poziția 0 cu 4 scuturi.  
Se adaugă planeta Venus pe poziția 1 cu 8 scuturi.  
Se adaugă planeta Earth pe poziția 2 cu 12 scuturi.  
Se adaugă planeta Jupiter pe poziția 3 cu 4 scuturi.  
Se adaugă planeta Saturn pe poziția 4 cu 12 scuturi.  
Se adaugă planeta Uranus pe poziția 5 cu 8 scuturi.  
Se adaugă planeta Neptunus pe poziția 6 cu 5 scuturi.  
Se adaugă planeta Pluto pe poziția 7 cu 9 scuturi.  
Planeta Neptun este înghițită de o gaură neagră.  
Planeta 2 (Earth) se rotește în sensul acelor de ceasornic cu 5 unități.  
Planeta 3 (Jupiter) se rotește în sens trigonometric cu 4 unități.  
se afișează informațiile despre planeta2 (Earth).  
Planeta 4 (Saturn) își consolidează scutul de pe poziția 2 cu 1 unitate.  
Planetele 3 (Jupiter) și 4 (Saturn) se ciocnesc.  
Se afișează informațiile despre planeta3 (Jupiter).  
Planeta 4 (Saturn) se rotește în sensul acelor de ceasornic cu 1 unitate.  
Planetele 3 (Jupiter) și 4 (Saturn) se ciocnesc. Planeta 3 (Jupiter) face implozie.  
Acum, planeta Earth va fi vecină cu planeta Saturn.  
Se afișează informațiile despre planeta Saturn.  
Comanda de show eșuază, deoarece avem 6 planete în galaxie, numerotate de la 0 la 5.  
Se adaugă planeta Tiaria pe poziția 4 în galaxie, între Saturn și Uranus.  
Se afișează informații despre aceasta.

## Precizări

- Datele se vor citi de la **intrarea standard (stdin)** și mesajele descrise în secțiunea **Input** se vor afișa la **ieșirea standard (stdout)**.
- În cazul în care o comandă generează un mesaj de eroare, comanda **nu se va efectua** și nu se vor afișa nici mesajele aferente. Ex. În cazul în care comanda ADD generează eroarea de tipul "Planet out of bounds!", nu se va mai afișa "The planet <planet> has joined the galaxy."

Implementarea galaxiei și a scuturilor planetelor trebuie realizată cu o listă circulară dublu înlănțuită generică. Nerespectarea acestui lucru va aduce o depunțare de 20p.

Pentru investigarea problemelor de tip *Segmentation Fault* sau comportament incorect al aplicației la unul din teste, pentru debugging, se recomandă folosirea gdb [<https://ocw.cs.pub.ro/courses/so/laboratoare/resurse/gdb>] Una dintre depunțări este pentru leak-uri de memorie [[https://en.wikipedia.org/wiki/Memory\\_leak](https://en.wikipedia.org/wiki/Memory_leak)]. În Linux pentru identificarea lor puteți folosi utilitarul valgrind [<http://valgrind.org/docs/manual/quick-start.html>].

Pentru instalarea `gdb` și `valgrind`, pe o distribuție Ubuntu se poate folosi comanda:

```
student@sd:~$ sudo apt-get install gdb valgrind
```

Pentru debugging și detectarea leak-urilor de memorie este necesar să ștergeți toate optimizările de la flag-urile de compilare (e.g. `-O3`) și trebuie să compilați doar cu flag-urile `-Wall -g` (sau cele care mai activează alte warning-uri, e.g. `-Wextra`).

Nu trebuie la fiecare eroare considerată fatală să eliberați fiecare pointer alocat dinamic. În cadrul corecturii temei principală verificare pentru memory leaks va fi pe o funcționare corectă/normală.

## TL; DR

---

- implementarea unei liste circulare dublu - înlănțuite generice.
- operații între planetele dintr-o galaxie:
  - add : adaugă o planetă.
  - black hole: elimină o anumită planetă.
  - upgrade shield: modifică valoarea unui scut.
  - expand shield: adaugă un scut la finalul listei de scuturi.
  - remove shield: elimină un anumit scut din lista de scuturi.
  - collide: două planete se ciocnesc.
  - rotate shield: rotirea liniei de scuturi a unei planete.
  - show info: afișarea informațiilor despre o anumită planetă.
- focusul temei este pe implementarea structurii de date CDLL generic.
- recomandăm crearea și testarea temeinică a CDLL (cu tot cu memory check), intrucat implementarea funcțiilor devine mult mai ușoară odată ce aveți o structură de date generică adaptată nevoilor voastre.

## Checker

---

Teste: 2021-tema1-check.zip.

Temele vor fi trimise pe vmchecker [<https://vmchecker.cs.pub.ro/ui/#SD>]. **Atenție!** Temele trebuie trimise în secțiunea **Structuri de Date (CA)**.

Arhiva trebuie să conțină:

- sursele **.c** și **.h**
- fișier **Makefile** cu două reguli:
  - regula **build**: în urma căreia se generează un executabil numit **main**
  - regula **clean** care șterge executabilul și fișierele obiect
- Compilarea trebuie să se realizeze cu flagurile **-Wall -Wextra -std=c99**
- fișier **README** care să conțină detalii despre implementarea temei

## Punctaj

---

**Atenție!** O temă care nu compilează pe vmchecker va primi 0 puncte.

1. 80p teste
2. **Fiecare** test este verificat cu valgrind. Dacă un test are memory leaks, nu va fi punctat.
3. 20p README + comentarii/claritate cod (**ATENȚIE!** Fișierul README trebuie făcut explicit, cât să se - înțeleagă ce ați făcut în sursă, dar fără comentarii inutile și detalii inutile).
4. Se acordă 20% din punctajul obținut pe teste, ca bonus pentru coding style. De exemplu, pentru o temă care obține maxim pe teste, se pot obține 20p bonus dacă nu aveți erori de coding style. Pentru o temă ce trece 18 teste din 20, se pot obține 18p dacă nu aveți erori de coding style.
5. O temă care obține 0p pe vmchecker este punctată cu 0.
6. Temele au deadline hard. Prin urmare, o temă trimisă după deadline este punctată cu 0.

**Nu copiați!** Toate soluțiile vor fi verificate folosind o unealtă de detectare a plagiatului. În cazul detectării unui astfel de caz, atât plagiatorul cât și autorul original (nu contează cine e) vor primi punctaj 0 pe toate temele! De aceea, vă sfătuim să nu vă lăsați rezolvări ale temelor pe calculatoare partajate (la laborator etc), pe mail/liste de discuții/grupuri etc.

## FAQ

---

**Q:** Se pot folosi flag-uri de optimizare?

**A:** Nu aveți voie să folosiți flag-uri de optimizare în Makefile (-O3, -O2, etc.).

**Q:** Tema 1 se poate face în C++?

**A:** Nu.

**Q:** Se pot folosi variabile globale?

**A:** Nu.

## Suport, întrebări și clarificări

---

Pentru întrebări sau nelămuriri legate de temă folosiți forumul temei [<https://curs.upb.ro/mod/forum/view.php?id=237904>]. Orice întrebare e recomandată să conțină o descriere cât mai clară a eventualei probleme. Întrebări de forma: "Nu merge X. De ce?" fără o descriere mai amănunțită vor primi un răspuns mai greu.

**ATENȚIE** să nu postați imagini cu părți din soluția voastră pe forumul pus la dispoziție sau orice alt canal public de comunicație. Dacă veți face acest lucru, vă asumați răspunderea dacă veți primi copiat pe temă.