

A trip throughout TV Series?

Responsabili:


- Alexandru-Gabriel Anica
- Andrei-Bogdan Florea
- George-Cristian Muraru

Termen de predare:

- Deadline soft: **15 Noiembrie 2020**
- Deadline hard: **22 Noiembrie 2020**

Pentru fiecare zi (24 de ore) de întârziere, se vor scădea 10 puncte din nota acordată, până la atingerea deadline-ului hard.

Întrebări

Dacă aveți nelămuriri, puteți să ne contactați pe forumul dedicat  temei de casă nr. 1 sau pe canalul Temei 1.

La orice întrebare vom răspunde în maxim 24 de ore.
Nu se acceptă întrebări în ultimele 24 de ore înainte de deadline.

Actualizări:

- 1.11.2020 - Full adder trebuie realizat cu porți logice
- 2.11.2020 - Fix checker pentru hunt - valori pentru sword human
- 3.11.2020 - Modificare DEBUG_LOG 11 -> 12
- 4.11.2020 - test_communication, mesaj mai explicit la DEBUG_LOG
- 5.11.2020 - Depunțare variabile globale
- 5.11.2020 - Makefile modificat - stergere fișiere task 3
- 12.11.2020 - Adaugat clarificare

Obiective Temă

- să se realizeze un program urmând anumite cerințe
- folosirea operatorilor pe biți
- să se însușească cunoștințele din primele trei laboratoare
- citirea și utilizarea codului scris de alte persoane

Cerință

Tema va fi împărțită în 3 părți independente, iar fiecare parte are diferite sarcini. Punctajul total pe teste este 90p, iar 10p se vor obține pe README.



Detalii legate de implementare se găsesc în scheletul de cod.

Part 1 - Bits & Pieces

Spock a fost trimis pe planeta Torothan datorită unei defecțiuni a teleportatorului. Acesta nu are echipamentul asupra sa și va trebui să folosească orice găsește prin deșert pentru a realiza un comunicator.

Pentru aceasta, ne cere ajutorul pentru a implementa câteva lucruri simple.

Codul pentru acest task trebuie scris în fișierul **src/gates.c**.

Part 1.1 - Bit by Bit

Având în vedere că Spock nu a mai lucrat de ceva timp cu biți, prima cerință este să implementăm următoarele operații pentru a-și reaminti

- selectarea unui bit
- flip-ul unui bit
- activarea unui bit
- negarea unui bit

Pentru mai multe detalii verificați TODO-urile din fișierul *gates.c* marcate cu Task 1.

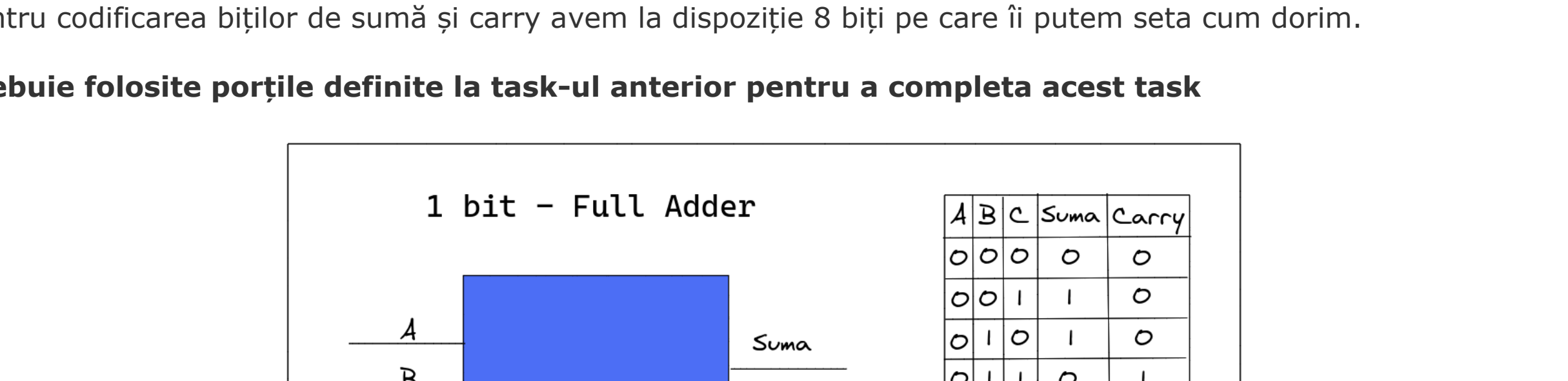
Part 1.2 - One gate to rule them all

În urma ajutorului primit de la cerința anterioară, acesta și-a amintit cum să construiască o poartă **nand**. Din păcate aceasta mai are nevoie și de alte porți logice pentru a realiza comunicatorul.

Pentru mai multe detalii verificați TODO-urile din fișierul *gates.c* marcate cu Task 2.

Utilizând doar **nand-uri** și porțile deja definite, trebuie să construim următoarele tipuri de porți: and, not, or si xor.

Porțile



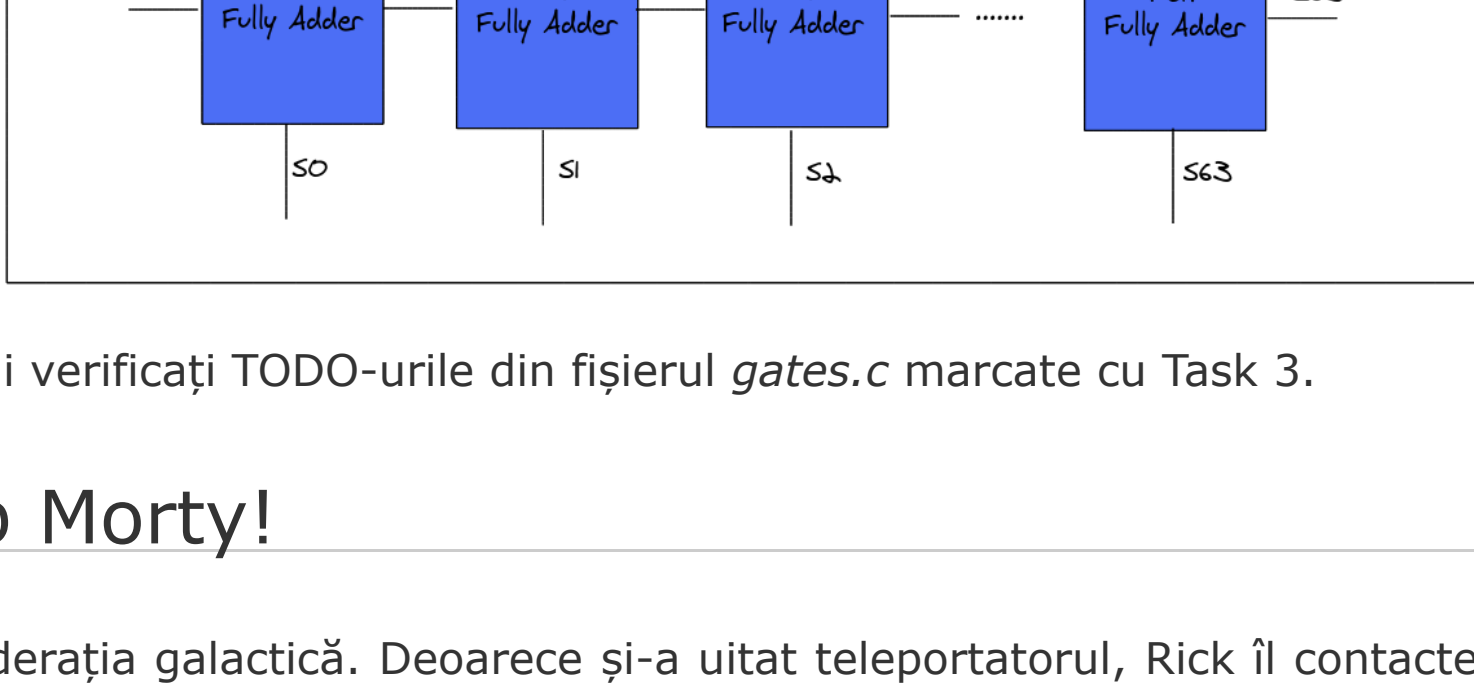
Part 1.3 - Just Carry the Bit

Pentru a realiza ultima parte din comunicator, Spock are nevoie de noi pentru a realiza un sumator pe biți. Vom implementa ceea ce se numește un **Ripple Carry Adder**.

Pentru a construi sumatorul vom avea nevoie de o componentă auxiliară **1 Bit Full Adder** care primește 3 biți la intrare și scrie 2 biți la ieșire, un bit de sumă și un bit de carry.

Pentru codificarea biților de sumă și carry avem la dispoziție 8 biți pe care îi putem seta cum dorim.

Trebuie folosite porțile definite la task-ul anterior pentru a completa acest task



Într-un final, pentru realizarea componentei finale (**Ripple Carry Adder**) vom utiliza **1 Bit Full Adder** ca în schema de mai jos.



Pentru mai multe detalii verificați TODO-urile din fișierul *gates.c* marcate cu Task 3.

Part 2 - Shut up Morty!

Rick a fost capturat de federația galactică. Deoarece și-a uitat teleportatorul, Rick îl contactează pe Morty pentru a-i oferi un set de instrucțiuni pentru trimiterea dispozitivului. Deoarece Morty nu prea se pricepe la programare, acesta apelează la noi pentru decodificarea mesajelor trimise de Rick.



Pentru printarea la standard output vom folosi:

```
fprintf(stdout, ...)
```

Codul pentru acest task trebuie scris în **src/communication.c**.

Part 2.1 - The Beginning

Rick i-a lăsat lui Morty 2 mecanisme prin care acesta poate primi și trimite un mesaj *send_squanch*, *recv_squanch*. Ambele primitive funcționează la nivel de octet:

- send_squanch* - trimite un octet
- recv_squanch* - recepționează un octet

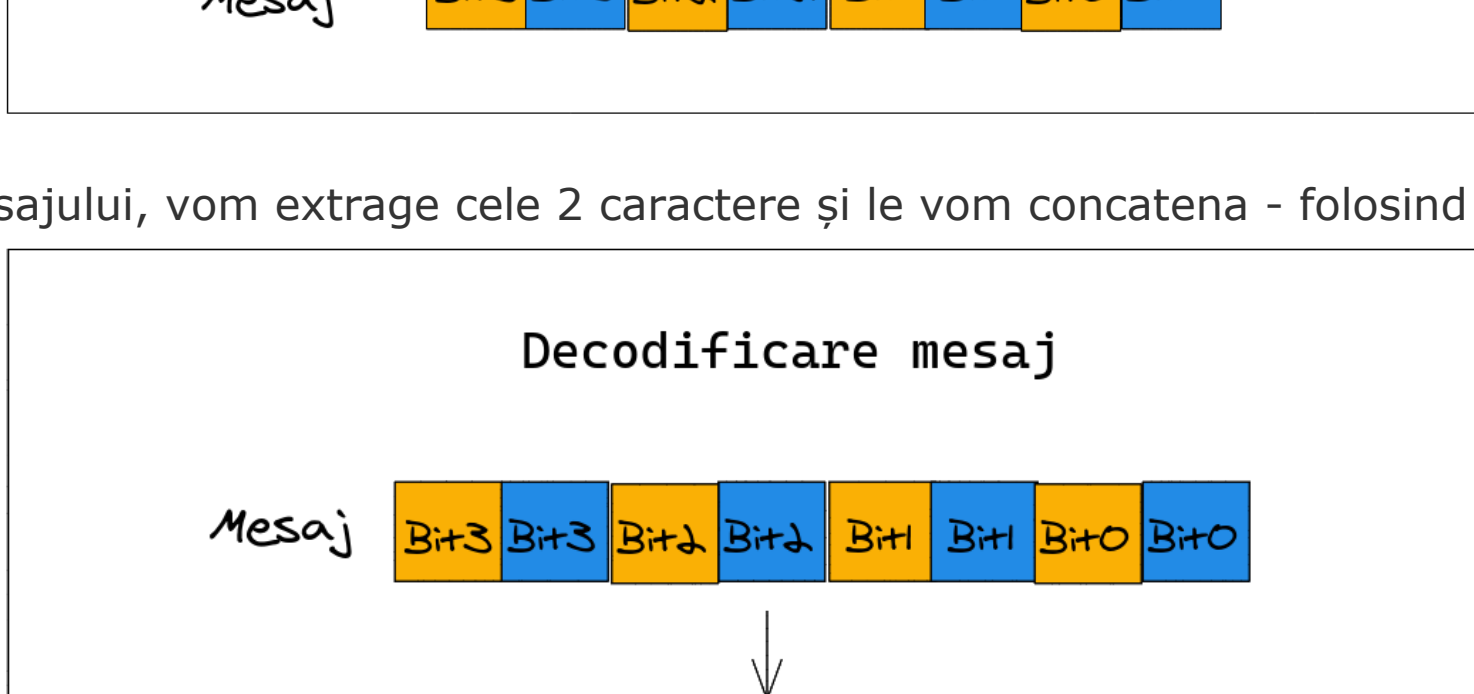
Pentru primul task, Morty ne cere să îl ajutăm să verifice dacă poate comunica cu Rick prin intermediul acestor 2 mecanisme. Un caracter este codificat prin numere de la 1 la 26: 1 este A, 2 este B, ..., 26 este Z.

Pentru mai multe detalii verificați TODO-urile din fișierul *communication.c* marcate cu Task 1.

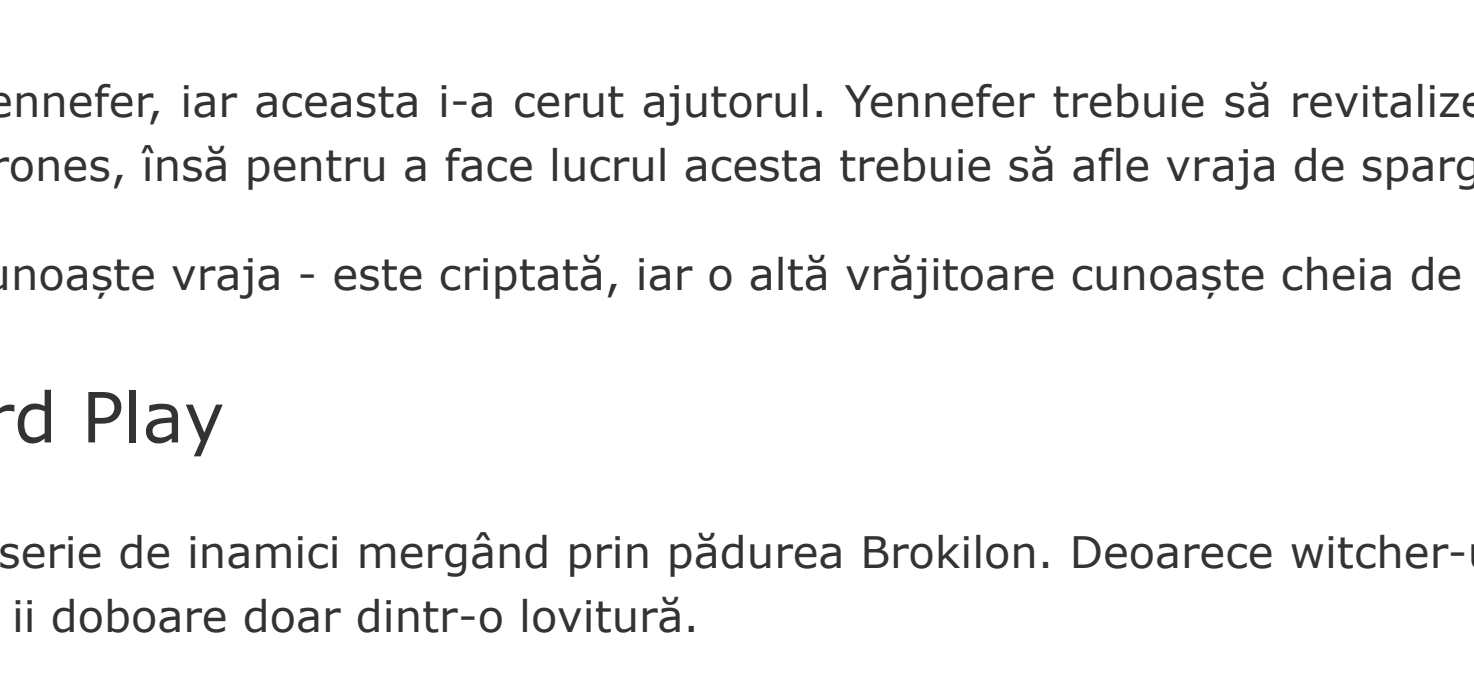
Part 2.2 - Waiting for the Message

Dupa ce ne-am obișnuit cu mecanismele de trimitere/primire de caractere, urmează să primim și să trimitem mesaje. Înainte de a primi/trimite mesajul trebuie să știm lungimea acestuia (numărul de caractere care vor fi trimise). Vom folosi *send_squanch/recv_squanch* (aceleași funcții) pentru a-l ajuta pe Morty să comunice cu Rick.

Lungimea mesajului va fi codificată pe un octet ca în imaginea de mai jos:



Pentru decodificare mesajului, vom extrage cele 2 caractere și le vom concatena - folosind *recv_squanch2*:



Pentru mai multe detalii verificați TODO-urile din fișierul *communication.c* marcate cu Task 3.

Part 3 - Toss a coin to your Witcher

În această parte îl vom urmări pe Geralt din Rivia și îl vom ajuta să își ducă la capăt quest-urile.

Codul pentru acest task trebuie scris în **src/hunt.c**.

Part 3.1 - Axii

Geralt s-a întâlnit cu Yennefer, iar aceasta i-a cerut ajutorul. Yennefer trebuie să revitalizeze o pădure care a fost pângărită de surorile Crones, însă pentru a face lucrul acesta trebuie să aștepte de spargere a blestemului.

Una dintre vrăjitoare cunoaște vraja - este cîmpită, iar o altă vrăjitoare cunoaște cheia de decriptare.

Part 3.3 - Sword Play


Geralt ține mereu 2 săbii cu el, una de argint pentru monștri și una de oțel pentru oameni. Fiecare personaj negativ care îl atacă pe Geralt are diferite puncte slabe. Pentru acest task trebuie să îl ajutăm pe Geralt să aleagă atât sabia corectă, cât și să impregneze un set de caracteristici sabiei alese.

Part 3.3 - The Witcher Trials

Pentru a ajunge un "witcher" o persoană trebuie să treacă prin mai multe etape. Aceste procese sunt defășurate în diferite "școli pentru witcheri", unde dintre acestea fiind Școala Lupului - aflată la Kaer Morhen.


Pentru acest task va trebui să oferim ajutorul nostru unui recrui pentru a trece prin aceste procese.

Trimitere temă

Tema va fi trimisă folosind  vmchecker, cursul **Programarea Calculatoarelor (CB & CD)**.

Găsiți arhiva cu checker-ul și makefile-ul  aici.

Formați arhivei va fi următorul:

- {gates.c, communication.c, hunt.c}
- Un fișier  README în care vă descrieți rezolvarea taskurilor.



- O parte din cerințe conțin explicații și în cod.
- Arhiva trebuie să fie de tip **zip**.
- Codul va fi scris doar în fișierele *gates.c*, *communication.c* și *hunt.c*
- Makefile-ul și testele vor fi cele din arhiva de test - noi vom dezarchiva arhiva voastră direct în folderul src
- În scheletul de cod veți vedea bucăți de cod care conțin assert-uri **assert(conditie)** - acestea sunt folosite pentru a valida inputul


Listă depunctări

Lista nu este exhaustivă.

- o temă care nu compilează și nu a rulat pe **vmchecker** nu va fi luată în considerare
- o temă care nu rezolvă cerința și trece testele pentru alte mijloace nu va fi luată în considerare
- [-1.0]: numele variabilelor nu sunt sugestive
- [-1.0]: linii mai lungi de 80 de caractere
- [-5.0]: abordare ineficientă
- [-10.0]: variabile globale
- [-100.0]: TOT punctajul, în cazul în care se încearcă "obținerea" punctajului pe temă folosind alte metode decât cele normale

- în cadrul cursului de programare nu avem ca obiectiv rezolvarea în cel mai eficient mod posibil a programelor; totuși, ne dorim ca abordarea să nu fie una ineficientă, de genul să nu folosiți instrucțiuni repetitive acolo unde clar nu era cazul, etc.

Resurse generale

- Anunțuri CB/CD
- Regulament: seria CA
- Regulament: seria CB/CD
-  Catalog 2021-2022 CB/CD
- Calendar

Tutoriale

- Good Practices
- Coding style example
- Debugging
- Read documentation
- Code understanding
- Logging
- Unit tests

Resurse laborator

- VM - CA
- Test practică - CA
- Checker laborator CB/CD
- WSL Setup

Cursuri

Continutul Tematic

Laboratoare

- Unele de programare
- Tipuri de date. Operatori.
- Instrucțiunile limbajului C
- Funcții
- Tablouri. Particularizare - vectori
- Matrice. Operații cu matrice
- Optimizarea programelor folosind operații pe biți
- Pointeri. Abordarea lucrului cu tablouri folosind pointeri
- Alocarea dinamică a memoriei. Aplicații folosind tablouri și matrice
- Prelucrarea șirurilor de caractere. Funcții. Aplicații
- Structuri. Uniuni. Aplicație: Matrice rare
- Operații cu fișiere. Aplicații folosind fișiere.
- Parametrii liniei de comandă. Preprocesorul. Funcții cu număr variabil de parametri
- Recapitulare

Teme de casă (general)

- Indicații generale

Teme de casă: seria CA

Teme de casă: seria CB/CD

- Tema 1
- Tema 2
- Tema 3

Table of Contents

- A trip throughout TV Series?
 - Responsabili:
 - Actualizări:
 - Obiective Temă
- Cerință
- Part 1 - Bits & Pieces
 - Part 1.1 - Bit by Bit
 - Part 1.2 - One gate to rule them all
 - Porțile
 - Part 1.3 - Just Carry the Bit
- Part 2 - Shut up Morty!
 - Part 2.1 - The Beginning
 - Part 2.2 - Waiting for the Message
 - Part 2.3 - In the Zone
- Part 3 - Toss a coin to your Witcher
 - Part 3.1 - Axii
 - Part 3.3 - Sword Play
 - Part 3.3 - The Witcher Trials
- Trimitere temă
 - Listă depunctări