

**VILNIAUS UNIVERSITETAS**  
**MATEMATIKOS IR INFORMATIKOS FAKULTETAS**  
**PROGRAMŲ SISTEMŲ KATEDRA**

**Sąsūkos kodas ir dekodavimas su grįžtamuju ryšiu**

Kodavimo teorijos pirma užduotis

Atliko:	4 kurso 6 grupės studentas	(parašas)
	Tomaš Monkevič	
Darbo vadovas:	asist. dr. Gintaras Skersys	(parašas)

Vilnius, 2018

## **TURINYS**

1. Programos paleidimas .....	3
2. Pradinių tekstų aprašymas .....	3
3. Vartotojo sąsajos aprašymas .....	3
4. Programiniai sprendimai .....	4
5. Eksperimentų aprašymai .....	4
5.1. Pirmas eksperimentas .....	4
5.2. Antras eksperimentas .....	5
5.3. Trečia eksperimentas .....	5

## 1. Programos paleidimas

Paleidžiamasis failas yra Release aplankale, pavadinimu CodingTheory.exe, kad paleisti programą užtenka tik du kartus greitai paspausti CodingTheory.exe failą. Jeigu programa nepasileis tai tame pačiame aplankale (Release) yra Microsoft Redistribution diegimo failas, kurį paleidus įsidsės reikalingos C++ bibliotekos ir programa turėtų veikti.

## 2. Pradinių tekstų aprašymas

*CodingTheory/Include/Channel.h* - realizuotas pranešimo siuntimas kanalu.

*CodingTheory/Include/Decoder.h* – realizuotas dekodavimas su grįžtamuju ryšiu.

*CodingTheory/Include/Encoder.h* – realizuotas sąsūkos kodas

*CodingTheory/Include/FiniteField.h* – baigtinio kūno realizacija

*CodingTheory/Include/Vector.h* – vektorius baigtinių kūnų

*CodingTheory/Include/Utils/Utils.h* – realizuota MDE funkcija

*CodingTheory/Source/Utils/stb\_image.cpp* – tai biblioteka kuri padeda skaityti BMP failo formatą.

*CodingTheory/Source/Utils/stb\_image\_write.cpp* – tai biblioteka kuri padeda įrašyti duomenys BMP formatu.

*CodingTheory/Source/Main.cpp* – realizuoti trys pagrindiniai scenarijai.

*CodingTheory/Source/Platforms/Linux/ColorPrinter.cpp* – realizuotas spalvotas išvedimas į konsolę Linux platformai.

*CodingTheory/Source/Platforms/Windows/WinColorPrinter.cpp* – realizuotas spalvotas išvedimas į konsolę Windows platformai.

## 3. Vartotojo sąsajos aprašymas

Vartotojo sąsaja yra padaryta su komandine eilute, todėl visas įvedimas yra vykdomas per klaviatūrą. Pagrindiniame menu įvedus skaičių nuo 1 iki 3 bus paleistas atitinkamas scenarijus, jeigu bus įvestas 0 programa baigs darbą, jeigu bus įvestas 4 galima pakeisti kanalo klaidos tikimybę.

Pasirinkus pirmą scenarijų iškart yra prašoma įvesti vektorių, kurių norime užkoduoti ir persiųsti per kanalą. Vektoriaus įvedimas vyksta rašant 0 arba 1, kai turimėsime norimą vektorių spaudžiame ENTER mygtuką ir vektorius bus užkoduotas ir pasiųstas kanalų.

Pasirinkus antrą scenarijų iškart yra prašoma įvesti tekstą. Tekstą įvedame spaudžiant ant klaviatūros, kad **baigti teksto įvedimą reikia naujoje eilutėje parašyti – „\n“ ir paspausti ENTER.**

Pasirinkus trečią scenarijų iškart yra prašoma įvesti paveiksluko vietą. Tekstas įvedame spaudžiant ant klaviatūros.

## 4. Programiniai sprendimai

Programa parašyta naudojant C++. BMP failo skaitymui ir rašymui yra naudojamos dvi bibliotekos: stb\_image ir stb\_image\_writer.

Vektoriaus siuntimas per kanalą yra realizuotas taip, kad iteruojant per kiekvieną baigtinį kūną vektoriuje yra generuojamas atsitiktinis skaičius ir jeigu tas sugeneruotas skaičius yra mažesnis už nurodytą kanalo klaidų tikimybę tai baigtinis kūnas yra apverčiamas (t.y. 0 į 1 arba 1 į 0).

Teksto ir paveiksluko užkodavimas, siuntimas kanalu ir dekodavimas yra vykdomas taip pat kaip su vektoriumi. Tiesiog prieš kodavimą tekstas arba baitai yra konvertuojami į vektorių.

## 5. Eksperimentų aprašymai

Žemiau yra aprašyti trys eksperimentai.

### 5.1. Pirmas eksperimentas

Iš nuotraukos apačioje, matome, kad kai yra padarytos mažiau negu trys klaidos 14 bitu intervale tai dekoderis gerai veikia.

```
Please enter vector: 1111000010100101
Encoded vector: 1111 1010 0100 0000 1101 1000 0110 0111 0101 0101 0101
Channel output: 1111 1010 0100 0000 1101 1000 0110 0111 0101 0101 0101
Errors made: 0

Do you want to eddit channel ouput? y/n: y
Enter positions to flip ('-1' - to finish):
> 0
> 4
> 29
> 36
> -1
Channel output after edit: 0111 0010 0100 0000 1101 1000 0110 0011 0101 1101 0101
Errors made: 4

Decoder output:
1111 0000 1010 0101
Errors made: 0
```

Taičiau jeigu bus padaryta viena klaida daugiau dekoderis blogai dekoduos ir dar propaguos klaidą į priki.

```
Please enter vector: 1111000010100101
Encoded vector: 1111 1010 0100 0000 1101 1000 0110 0111 0101 0101 0101
Channel output: 1111 1010 0100 0000 1101 1000 0110 0111 0101 0101 0101
Errors made: 0

Do you want to eddit channel ouput? y/n: y
Enter positions to flip ('-1' - to finish):
> 0
> 4
> 8
> 29
> 36
> -1
Channel output after edit: 0111 0010 1100 0000 1101 1000 0110 0011 0101 1101 0101
Errors made: 5

Decoder output:
0100 1000 1010 0101
Errors made: 4
```

## 5.2. Antras eksperimentas

Iš nuotraukos apačioje, matome, kad dekoderis gan gerai susitvarkė ir paliko tik 9 klaidas, kur tekstas siųstas be kodavimo padarė 48 klaidas, kai kanalo klaidos tikimybė buvo 5%.

```
Enter text (\q - to finish):
> Laba diena,
>
> Rasau testini tekstą, kad galima būtų patikrinti feedback dekodavimą.
> Tikekimes kad veiks :)
>
> Pagarbiai,
> Tomas Monkevic
> \q
--- with encoding ---
106 of 2140 errors made in channel.
-----Output-----
Laba diena,

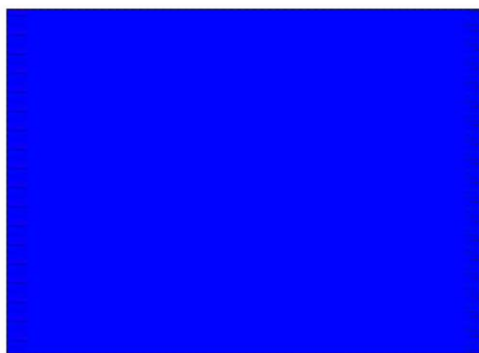
Rasau testini tekstą, kad galima būtų Čluirinti feedback dekodavimą.
Tikekkimes kadiveiczA:)

Pagarbiai,
Tomas Monkevic
-----
Errors made: 9

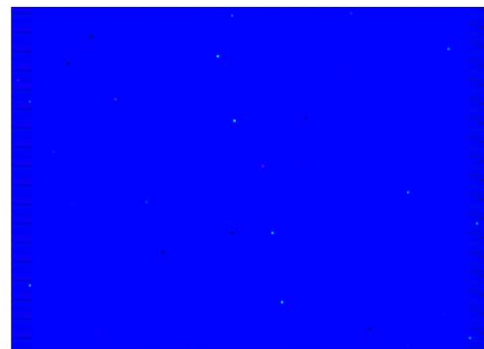
--- without encoding ---
60 of 1064 errors made in channel.
-----Output-----
RbC diela(Q
Ras!q`testhnaEt0kSta, cad Calimc `ttu$ptiksonti feedbaao dekodavimc.
Dikekjihe5`k`d teyIsA:)*(Pagarbiak,JToeIs iOnou~ic
-----
Errors made: 48
-----Menu-----
```

## 5.3. Trečia eksperimentas

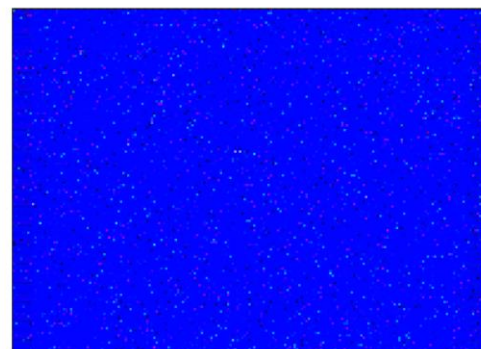
Kai kanalo klaidos tikimybė buvo 1% kodavimas beveik ištaisė visas klaidas nuotraukoje. Nuotraukos dydis buvo 84,4 KB



Originalus



Su kodavimu



Be kodavimo