Matveeva Lada Ivanna AIS ID : 99198 Lab 10

Task 1. (ex1.py)

A1 = 00 A2 = 01 A3 = 10 A4 = 11 A5 = 100 A6 = 101

M = A1 A2 A3 A4 A5 A6
00 01 10 11 100 101

Redundant message M

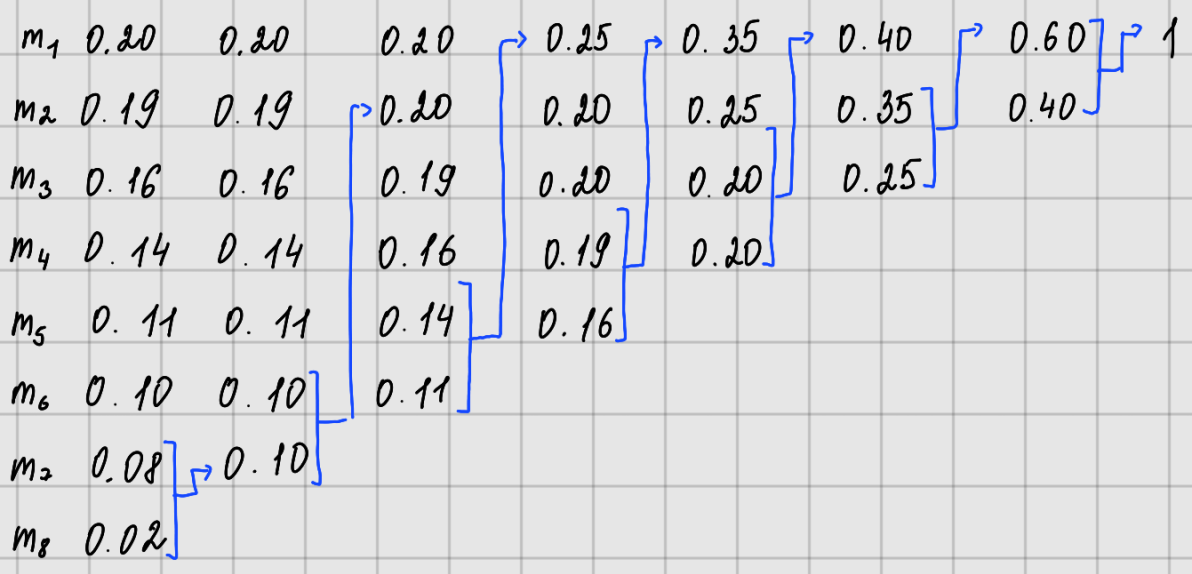
A1 A2 A3 A4 A5 A6 =
000 001 101 110 1001 1010

A1 = 000 0 + 0 = 0 0 mod 2 = 0
A2 = 011 0 + 1 = 1 1 mod 2 = 1
A3 = 101 1 + 0 = 1 1 mod 2 = 1
A4 = 110 1 + 1 = 2 2 mod 2 = 0
A5 = 1001 1 + 0 + 0 = 1 1 mod 2 = 1
A6 = 1010 1 + 0 + 1 = 2 2 mod 2 = 0

Results from ex1.py

```
Redundant message M:  ['000', '011', '101', '110', '1001', '1010']
```

Task 2 (ex2.py)



Results from ex2.py

	0	1	2	3	4	5	6	7
0	0.20	0.20	0.20	0.25	0.35	0.40	0.60	1.00
1	0.19	0.19	0.20	0.20	0.25	0.35	0.40	
2	0.16	0.16	0.19	0.20	0.20	0.25		
3	0.14	0.14	0.16	0.19	0.20			
4	0.11	0.11	0.14	0.16				
5	0.10	0.10	0.11					
6	0.08	0.10						
7	0.02							

Task 3

Redundant codes are useful in cryptographic systems to enhance error detection and correction, making them valuable for secure communication over unreliable channels. They improve the stability of information transmission by allowing the receiver to identify and correct errors, ensuring data integrity. Redundancy is crucial for scenarios where stable transmission is essential, like in noisy environments or long-distance communications.

Non-redundant codes, on the other hand, are appropriate where efficiency is prioritized, such as in environments with low error probability or when computational overhead needs to be minimized. They are best used when the communication channel is reliable, and data integrity checks are either unnecessary or handled by other means.