Кол-во баллов: 12
Дедлайн 1: 27 мая, дедлайн 2: 3 июня (7 баллов)

вариант-1

1. Для заданного многочлена $x^n - 1$ над полем $\mathbb{F}_p$ найти

i) количество циклических кодов длины n над полем $\mathbb{F}_p$;

ii) порождающие и проверочные многочлены для этих кодов (если кодов больше 6, то только для 6 из них);

iii) размерность каждого кода из п.2;

iv) порождающие матрицы любых двух нетривиальных кодов из п. 2;

v) записать порождающие матрицы кодов из прошлого пункта в систематическом виде.

Пусть v — ваш номер в списке группы. Тогда n и p выбрать из соответствующей строки таблицы:

$v \mod 11$	0	1	2	3	4	5	6	7	8	9	10
n	17	10	13	23	20	14	15	11	11	27	13
p	2	3	5	2	3	5	2	3	5	2	3

2. Для любого из кодов из задания 1 построить таблицу Мэггита для ошибок веса 1 и 2, состоящих только из единиц и нулей.

3. Программно реализовать алгоритм сжатия Хаффмана или арифметическое кодирование для произвольных строк. Сравнить среднее количество битов на символ в сжатой строке со значением энтропии.

вариант-2

1. Реализовать функцию, которая принимает на вход число n и проверочный многочлен $h(x) \in \mathbb{F}_2[x]/(x^n - 1)$ и возвращает число ошибок, которое может гарантировано исправить одноитерационная версия декодера *shift-sum*. Напомним, что это число может быть найдено следующим образом:

i) пусть $\hbar(y) = \sum_{i=0}^{n-1} h_i y^i \in \mathbb{Z}[y]/(y^n - 1)$ — целочисленная версия многочлена h, пусть

$$\hbar^*(y) = \hbar(y^{-1}) = \sum_{i=0}^{n-1} h_{n-i} y^i.$$

На первом шаге необходимо вычислить многочлен $u(y) = \hbar(y) \cdot \hbar^*(y) \mod y^n - 1$ *(важно: умножение не над полем $\mathbb{F}_2$, а над $\mathbb{Z}$!).*

ii) через $\mu(w)$ обозначим сумму w наибольших несвободных *(т.е. стоящих перед ненулевыми степенями y)* коэффициентов многочлена $u(y)$. Тогда гарантированная корректирующая способность декодера может быть найдена как наибольшее натуральное число t, удовлетворяющее неравенству

$$\mu(t) + \mu(t-1) < \mathrm{wt}(h)$$

2. При заданных n и q можно определить множество

$$C_s = \{s \cdot q^i \mod n \mid i \in \mathbb{N}\}$$

(отметим, что начиная с некоторого i последовательность $sq^i \mod n$ зациклится) называемое циклотомическим классом числа s. Если $q = 2$, то каждому циклотомическому классу можно поставить в соответствие следующий многочлен

$$h_s(x) = \sum_{i \in C_s} x^i$$

из кольца $\mathbb{F}_2[x]/(x^n - 1)$.

Пусть v — ваш номер в списке группы. Выберите число n из следующей таблицы

$v \mod 8$	0	1	2	3	4	5	6	7
n	73	127	819	255	273	1023	117	63

и найдите все циклотомические классы C_s при $q = 2$.

```
# SageMath
Zmod(n).cyclotomic_cosets(q=2)
```

Для каждого C_s найдите многочлен $h_s(x)$ и примените к нему функцию из задания 1.

вариант-3

1. Программно реализовать итеративную версию декодера *shift-sum.* Напомним, что этот декодер работает следующим образом:

i) на вход подаётся проверочный многочлен $h(x) \in \mathbb{F}_2[x]/(x^n - 1)$ и зашумлённое кодовое слово $z(x) = c(x) + e(x)$

ii) вычисляется синдром $s(x) = z(x)h(x)$. Если $s(x) = 0$, то алгоритм останавливается и возвращает $z(x)$.

iii) если $s(x) \neq 0$, то строится целочисленная версия многочлена $s(x)$

$$\mathscr{s}(y) = \sum_{i=0}^{n-1} s_i y^i \in \mathbb{Z}[y]/(y^n - 1)$$

(т.е. просто забываем то, что $s(x)$ — многочлен над конечным полем). Затем вычисляется многочлен

$$\phi(y) = \mathscr{s}(y) \cdot \mathscr{h}^*(y),$$

где $\mathscr{h}^*(y)$ определён так же как в варианте 2.

iv) Коэффициенты многочлена $\phi(y)$ определяют «надёжность» каждой координаты *(чем больше ϕ_i — тем выше вероятность того, что в i–ой координате ошибка).* Поэтому можно попробовать исправить ошибку в $z(x)$ следующим образом:

$$z(x) := z(x) - \sum_{\phi_i = \phi_{max}} x^i,$$

где $\phi_{max} = \max_i\{\phi_i\}$.

После обновления $z(x)$, если не превышено максимальное число итераций, происходит переход обратно к шагу *ii).*

2. Протестируйте работу декодера на одном из следующих кодов

$$q = 2, \quad n = 21, \quad h(x) = h_7(x) + h_9(x)$$

$$q = 2, \quad n = 255, \quad h(x) = h_1(x) + h_{27}(x)$$

$$q = 2, \quad n = 73, \quad h(x) = h_1(x)$$

(см. определение h_s в варианте 2). В качестве $c(x)$ используйте нулевое кодовое слово, а ошибку генерируйте случайно. При каких весах ошибки вероятность корректного декодирования не меньше 0.9?

Замечание. *Вычисление синдрома $s(x)$ может быть реализовано следующим образом:*

$$s(x) = z(x)h(x) = \left(\bigoplus_{i=0}^{n-1} z_i x^i\right) \cdot \left(\bigoplus_{i=0}^{n-1} h_i x^i\right) = \bigoplus_{i=0}^{n-1} \left(\bigoplus_{j=0}^{n-1} z_{(i-j)\%n} h_j\right) x^i,$$

а вычисление $\phi(y)$ следующим образом:

$$\phi(x) = \sum_{i=0}^{n-1} \left(\sum_{j=0}^{n-1} s_{(i+j)\%n} h_j\right) x^i.$$

Можно также использовать готовые реализации факторколец в SageMath.

```
# SageMath
P.<z> = GF(2)[]  # кольцо многочленов над полем F_2
R.<x> = P.quo(z^n-1)  # кольцо F_2[x]/(x^n - 1)
Q.<a> = ZZ[]  # целочисленные многочлены
K.<y> = Q.quo(a^n-1)  # Z[y]/(y^n - 1)

```

3. Программно реализовать алгоритм сжатия Хаффмана.