Задача A. Алгоритм Витерби мягкого декодирования линейных блоковых кодов

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать кодер линейного блокового кода и его декодер на основе алгоритма Витерби. Исходными данными являются длина n, размерность k и порождающая матрица G двоичного линейного блокового кода.

Формат входных данных

файл input.txt должен начинаться следующим образом:

```
n k
G
```

Далее должны быть представлены команды.

Формат выходных данных

Первой строкой в файле output.txt должна быть последовательность $|V_i|, 0 \leqslant i \leqslant n$, где $|V_i|$ — число узлов в решетке на ярусе i. Далее должны быть приведены результаты выполнения команд. Моделирование следует производить для случая канала с двоичной амплитудно-импульсной модуляцией и аддитивным белым гауссовским шумом. Под уровнем шума следует понимать отношение сигнал/шум на бит, выраженное в децибелах.

Пример

input.txt	output.txt
8 4 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 Encode 1 0 0 0 Decode -1.0 1.0 1 1 1 1 1 1.5 Simulate 3 100000 100 Simulate 4 100000 100	1 2 4 8 4 8 4 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 2.56E-2 9.31E-3

Задача B. Алгоритм Витерби мягкого декодирования сверточных кодов

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать несистематический кодер сверточного кода со скоростью $1/2$ и его декодер на основе алгоритма Витерби. После обработки заданной информационной последовательности кодер должен быть переведен в нулевое состояние путем принудительной подачи на его вход K нулевых битов, где K — длина кодового ограничения. Моделирование следует производить для случая канала с двоичной амплитудно-импульсной модуляцией и аддитивным белым гауссовским шумом. Под уровнем шума следует понимать отношение сигнал/шум на бит, выраженное в децибелах.

Формат входных данных

Первая строка в файле input.txt имеет вид

`G0 G1 k`

Здесь $G0, G1$ — порождающие многочлены кода, представленные в виде битовых масок в восьмеричной системе счисления, k — число кодируемых информационных символов (в десятичной системе счисления).

Формат выходных данных

Первая строка выходного файла должна содержать минимальное расстояние полученного блокового кода. Далее в выходном файле должны быть приведены результаты выполнения команд. Моделирование следует производить для случая канала с двоичной амплитудно-импульсной модуляцией и аддитивным белым гауссовским шумом. Под уровнем шума следует понимать отношение сигнал/шум на бит, выраженное в децибелах.

Пример

input.txt
023 037 6 Encode 1 0 0 1 0 1 Decode -1 1.2 -1 -1 1 -1 -1 1 1 1 -1 1 -1 1 -1 1 1 -1 -1 -1 Simulate 4 100000 100
output.txt
6 1 1 1 1 0 1 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1.02E-2

Задача C. Алгоритм Сугиямы декодирования двоичных кодов Гоппы

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать процедуру построения проверочной матрицы двоичного кода Гоппы длины $n = 2^m$, его систематическое кодирование и декодирование в метрике Хемминга. Для решения ключевого уравнения следует использовать алгоритм Сугиямы (расширенный алгоритм Евклида).

Под уровнем шума следует понимать вероятность ошибки на бит в двоичном симметричном канале. В качестве локаторов кода при длиной 2^m следует рассматривать все элементы конечного поля в последовательности $0, \alpha^0, \alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^{2^m-2}$. Для кодов длиной $2^m - 1$ следует исключить нулевой локатор. Гарантируется, что многочлен Гоппы свободен от квадратов и не имеет корней в $GF(2^m)$.

При построении порождающей матрицы кода в систематическом виде следует обеспечить размещение единичной матрицы в позициях с наименьшими возможными номерами.

Формат входных данных

В первой строке файла содержится длина кода 2^m или $2^m - 1$, примитивный многочлен поля $GF(2^m)$ (представленный в виде битовой маски в десятичной систем счисления) и степень многочлена Гоппы. Во второй строке выписаны коэффициенты многочлена Гоппы начиная с нулевого в виде чисел в десятичной системе счисления, двоичное представление которых соответствует их разложению по стандартному базису $GF(2^m)$. Далее следуют команды.

Формат выходных данных

В первой строке выходного файла должна быть выписана размерность кода. Далее должны быть приведены результаты выполнения команд

Пример

input.txt	output.txt
32 37 2 1 1 1 Encode 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Decode 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Simulate 0.1 100000 100	22 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0100000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.6330164805 0 1 1 0 1 1 1 1 1

Задача D. Алгоритм Берлекэмпа-Месси декодирования двоичных кодов Гоппы

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать процедуру построения проверочной матрицы двоичного кода Гоппы длины $n = 2^m$, его систематическое кодирование и декодирование в метрике Хемминга. Для решения ключевого уравнения следует использовать алгоритм Берлекэмпа-Месси. Под уровнем шума следует понимать вероятность ошибки на бит в двоичном симметричном канале. В качестве локаторов кода при длиной 2^m следует рассматривать все элементы конечного поля в последовательности $0, \alpha^0, \alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^{2^m-2}$. Для кодов длиной $2^m - 1$ следует исключить нулевой локатор. Гарантируется, что многочлен Гоппы свободен от квадратов и не имеет корней в $GF(2^m)$.

Формат входных данных

В первой строке файла содержится длина кода 2^m или $2^m - 1$, примитивный многочлен поля $GF(2^m)$ (представленный в виде битовой маски в десятичной систем счисления) и степень многочлена Гоппы. Во второй строке выписаны коэффициенты многочлена Гоппы начиная с нулевого в виде чисел в десятичной системе счисления, двоичное представление которых соответствует их разложению по стандартному базису $GF(2^m)$. Далее следуют команды.

Формат выходных данных

В первой строке выходного файла должна быть выписана размерность кода. При построении порождающей матрицы кода в систематическом виде следует обеспечить размещение единичной матрицы в позициях с наименьшими возможными номерами.

Пример

input.txt	output.txt
32 37 2 1 1 1 Encode 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Decode 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Simulate 0.1 100000 100	22 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0100000000000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6330164805 0 1 1 0 1 1 1 1 1

Задача E. Последовательное декодирование полярных кодов

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	1024 мегабайта

Необходимо реализовать процедуру построения полярных кодов для случая двоичного стирающего канала, их несистематическое кодирование, моделирование передачи и декодирование с помощью последовательного алгоритма. При реализации следует полагать максимальное число путей равным $D = kL$, где k — размерность кода, L — максимальное число проходов через одну фазу.

Кодирование должно осуществляться как $c = u \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes m}$, где $u_i = 0, i \in \mathcal{F}$, $\mathcal{F}$ — множество номеров замороженных символов. Прочие элементы вектора u представляют собой последовательно выбираемые элементы кодируемого информационного вектора. Моделирование следует производить для случая канала с двоичной амплитудно-импульсной модуляцией и аддитивным белым гауссовским шумом. Под уровнем шума следует понимать отношение сигнал/шум на бит, выраженное в децибелах.

Формат входных данных

Входной файл на первой строке содержит длину кода 2^m, размерность k, целевую вероятность стирания в двоичном стирающем канале и размер списка в декодере Тала-Варди. Далее идут строки с командами.

Формат выходных данных

Первая строка выходного файла должна содержать список номеров замороженных символов (нумерация с 0), выписанных в порядке возрастания. Далее должны быть представлены результаты выполнения команд

Пример

input.txt	output.txt
8 4 0.5 16 Encode 1 0 0 0 Decode -0.5 1 1 1 -1 -1 -1 -1 Simulate 3 100000 100 Simulate 4 100000 100	0 1 2 4 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 2.56E-2 9.31E-3

Задача F. Алгоритм Сугиямы декодирования двоичных кодов Рида-Соломона

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать процедуру построения примитивного кода Рида-Соломона в узком смысле над $GF(2^m)$, его систематическое кодирование и декодирование в метрике Хемминга с помощью расширенного алгоритма Евклида. Под уровнем шума следует понимать вероятность ошибки на символ в 2^m-ичном симметричном канале. Ненулевые символы вектора ошибки должны принимать значения из $GF(2^m) \setminus \{0\}$ с одинаковой вероятностью.

Формат входных данных

Файл input.txt должен содержать в первой строке длину кода $n = 2^m - 1$, примитивный многочлен поля $GF(2^m)$ (представленный в виде битовой маски в десятичной систем счисления) и конструктивное расстояние. Далее должны быть представлены команды.

Формат выходных данных

В первой строке выходного файла должна быть приведена размерность полученного кода k. Далее должен быть приведен порождающий многочлен кода $g(x) = \sum_{i=0}^{n-k} g_i x^i$ в виде последовательности его коэффициентов $g_0, g_1, \ldots, g_{n-k}$. Каждый коэффициент должен быть представлен в виде числа в десятичной системе счисления, двоичное представление которого соответствует его разложению по стандартному базису $GF(2^m)$. Далее должны быть представлены результаты выполнения команд.

Пример

input.txt	output.txt
7 11 3 Encode 1 0 0 0 0 Decode 3 6 1 0 0 0 1 Simulate 0.1 100000 100	5 3 6 1 3 6 1 0 0 0 0 3 6 1 0 0 0 0 0.15

Задача G. Алгоритм Берлекэмпа-Месси декодирования кодов Рида-Соломона

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать процедуру построения примитивного кода Рида-Соломона в узком смысле над $GF(2^m)$, его систематическое кодирование и декодирование в метрике Хемминга с помощью алгоритма Берлекэмпа-Месси. Под уровнем шума следует понимать вероятность ошибки на символ в 2^m-ичном симметричном канале. Ненулевые символы вектора ошибки должны принимать значения из $GF(2^m) \setminus \{0\}$ с одинаковой вероятностью.

Формат входных данных

Файл input.txt должен содержать в первой строке длину кода $n = 2^m - 1$, примитивный многочлен поля $GF(2^m)$ (представленный в виде битовой маски в десятичной систем счисления) и конструктивное расстояние. Далее должны быть представлены команды.

Формат выходных данных

В первой строке выходного файла должна быть приведена размерность полученного кода k. Далее должен быть приведен порождающий многочлен кода $g(x) = \sum_{i=0}^{n-k} g_i x^i$ в виде последовательности его коэффициентов $g_0, g_1, \ldots, g_{n-k}$. Каждый коэффициент должен быть представлен в виде числа в десятичной системе счисления, двоичное представление которого соответствует его разложению по стандартному базису $GF(2^m)$. Далее должны быть представлены результаты выполнения команд.

Пример

input.txt	output.txt
7 11 3 Encode 1 0 0 0 0 Decode 3 6 1 0 0 0 1 Simulate 0.1 100000 100	5 3 6 1 3 6 1 0 0 0 0 3 6 1 0 0 0 0 0.15

Задача H. Списочное декодирование полярных кодов

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать процедуру построения полярных кодов для случая двоичного стирающего канала, их несистематическое кодирование, моделирование передачи и декодирование с помощью min-sum версии алгоритма Тала-Варди. Кодирование должно осуществляться как $c = u \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes m}$, где $u_i = 0, i \in \mathcal{F}$, $\mathcal{F}$ — множество номеров замороженных символов. Прочие элементы вектора u представляют собой последовательно выбираемые элементы кодируемого информационного вектора. Моделирование следует производить для случая канала с двоичной амплитудно-импульсной модуляцией и аддитивным белым гауссовским шумом. Под уровнем шума следует понимать отношение сигнал/шум на бит, выраженное в децибелах. Из списка, формируемого декодером Тала-Варди, необходимо выбирать наиболее вероятное кодовое слово.

Формат входных данных

Входной файл на первой строке содержит длину кода 2^m, размерность k, целевую вероятность стирания в двоичном стирающем канале и размер списка в декодере Тала-Варди. Далее идут строки с командами.

Формат выходных данных

Первая строка выходного файла должна содержать список номеров замороженных символов (нумерация с 0), выписанных в порядке возрастания. На последующих строках должны быть приведены результаты выполнения команд

Пример

input.txt	output.txt
8 4 0.5 16 Encode 1 0 0 0 Decode -0.5 1 1 1 -1 -1 -1 -1 Simulate 3 100000 100 Simulate 4 100000 100	0 1 2 4 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 2.56E-2 9.31E-3

Задача I. Алгоритм Сугиямы (Евклида) декодирования двоичных кодов БЧХ

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать процедуру построения двоичного примитивного кода БЧХ в узком смысле, его систематическое кодирование и декодирование в метрике Хемминга с помощью расширенного алгоритма Евклида.

Формат входных данных

Файл input.txt должен содержать в первой строке длину кода $n = 2^m - 1$, примитивный многочлен поля $GF(2^m)$ (представленный в виде битовой маски в десятичной систем счисления) и конструктивное расстояние. Далее должны быть представлены команды. Под уровнем шума следует понимать вероятность ошибки на бит в двоичном симметричном канале.

Формат выходных данных

В первой строке выходного файла должна быть приведена размерность полученного кода k. На следующей строке должен быть приведен порождающий многочлен кода $g(x) = \sum_{i=0}^{n-k} g_i x^i$ в виде последовательности его коэффициентов $g_0, g_1, \ldots, g_{n-k}$. Далее должны быть представлены результаты выполнения команд.

Пример

input.txt	output.txt
7 11 3 Encode 1 0 0 0 Decode 0 1 0 1 0 0 0 Simulate 0.1 100000 100	4 1 1 0 1 1 1 0 1 0 0 0 1 1 0 1 0 0 0 0.15

Задача J. Алгоритм Берлекэмпа-Месси декодирования двоичных кодов БЧХ

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Необходимо реализовать процедуру построения двоичного примитивного кода БЧХ в узком смысле, его систематическое кодирование и декодирование в метрике Хемминга с помощью алгоритма Берлекэмпа-Месси.

Формат входных данных

Файл input.txt должен содержать в первой строке длину кода $n = 2^m - 1$, примитивный многочлен поля $GF(2^m)$ (представленный в виде битовой маски в десятичной систем счисления) и конструктивное расстояние. Далее должны быть представлены команды. Под уровнем шума следует понимать вероятность ошибки на бит в двоичном симметричном канале.

Формат выходных данных

В первой строке выходного файла должна быть приведена размерность полученного кода k. На следующей строке должен быть приведен порождающий многочлен кода $g(x) = \sum_{i=0}^{n-k} g_i x^i$ в виде последовательности его коэффициентов $g_0, g_1, \ldots, g_{n-k}$. Далее должны быть представлены результаты выполнения команд.

Пример

input.txt	output.txt
7 11 3 Encode 1 0 0 0 Decode 0 1 0 1 0 0 0 Simulate 0.1 100000 100	4 1 1 0 1 1 1 0 1 0 0 0 1 1 0 1 0 0 0 0.15

Задача K. Алгоритм рекурсивного мягкого декодирования по решеткам

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	1024 мегабайта

Необходимо реализовать кодер линейного блокового кода и его декодер на основе алгоритма Фудзивары-Ямамото-Касами-Линя (включая процедуру поиска оптимального секционирования).

Моделирование следует производить для случая канала с двоичной амплитудно-импульсной модуляцией и аддитивным белым гауссовским шумом. Под уровнем шума следует понимать отношение сигнал/шум на бит, выраженное в децибелах.

Формат входных данных

Исходными данными являются длина n, размерность k и порождающая матрица G двоичного линейного блокового кода. Таким образом, файл input.txt должен начинаться следующим образом:

```
n k
G
```

Формат выходных данных

В первой строке выходного файла должно быть указано число операций сложения и сравнения, выполняемых декодером. В последующих строках должны быть приведены результаты выполнения команд.

Пример

input.txt	output.txt
8 4 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 Encode 1 0 0 0 Decode -1.0 1.0 1 1 1 1 1 1.5 Simulate 3 100000 100	23 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 2.56E-2

Задача L. Алгоритм box-and-match декодирования линейных блоковых кодов

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	15 секунд
Ограничение по памяти:	1024 мегабайта

Необходимо реализовать кодер линейного блокового кода и его декодер на основе алгоритма box&match. Исходными данными являются длина n, размерность k и порождающая матрица G двоичного линейного блокового кода, а также параметры алгоритма "порядок переработки"(reprocessing order) t и длина контрольной полосы (control band) s.

Формат входных данных

файл input.txt должен начинаться следующим образом:

```
n k
G
t s
```

Далее в файле представлены команды.

Формат выходных данных

В выходном файле должны быть приведены результаты выполнения команд.

Пример

input.txt	output.txt
8 4 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 1 2 Encode 1 0 0 0 Decode -1.0 1.0 1 1 1 1 1 1.5 Simulate 6 100000 100	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 4.4E-4