Программа работает по следующему алгоритму:

1. Считываются входные данные и считаются вероятности символов (один набор вероятностей для простого варианта кода, и N наборов для контекстного, где N – количество различных символов во входных данных)
2. На основе полученных вероятностей символов производиться кодирование и запись входного файла.
3. Производится чтение закодированных данных и их декодирование.

В качестве тестовых файлов использовались:

1. В качестве тестового файла использовался файл, сгенерированный с помощью марковской цепи, описываемой следующей матрицей переходных вероятностей:

[0.743278 0.0278329 0.0559099 0.131626 0.0413525;

0.847072 0.0146794 0.126469 0.000161 0.0117801;

0.0946684 0.258797 0.232765 0.14655 0.26722;

0.125095 0.0801111 0.0470901 0.0587176 0.688986;

0.336406 0.398602 0.259163 0.00433363 0.00149548];

2. файл data\_static.txt, сгенерированный на основе стационарного распределения вероятностей

[0 0.308594 0.21875 0.0703125 0.078125 0.0507813 0.046875 0.00390625 0.00390625 0 0]

**Результаты сжатия дискретного стационарного источника без памяти**

Изначальный вес файла = 200 байт

Вес закодированного файла при верном распределении = 57 байт

Вес закодированного файла при неверном распределении [0 0.00390625 0.00390625 0.046875 0.078125 0.0507813 0.0703125 0.21875 0.308594 0 0] = 159 байт

Вес закодированного файла при верном распределении при использовании контекстного варианта = 46 байт.

При сжатии данных обладающих неравномерным распределением, арифметический код демонстрирует более высокую степень сжатия по сравнению с кодом Хаффмана.