МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность Информационные системы и технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №11 НА ТЕМУ:**

**Сжатие/распаковка данных арифметическим методом**

Выполнил студент 3 курса 1 группы

Халалеенко Андрей Николаевич

Минск 2022

**Задание 1:** Разработать авторское приложение в соответствии с целью лабораторной работы.

**Задание 2:** С помощью приложения выполнить прямое и обратное преобразования сообщений в соответствии с таблицей.

Для выполнения этого задания был использован следующий код на языке *python*:

def get\_intervals(probs):

    intervals = []

    lower = 0

    upper = 0

    for key, value in probs.items():

        upper += value

        intervals.append((key, lower, upper))

        lower += value

    return intervals

def compress(seq, probs):

    lower = 0

    higher = 1

    intervals = get\_intervals(probs)

    for character in seq:

        current\_range = higher - lower

        for element in intervals:

            if(character == element[0]):

                char\_upper = element[2]

                char\_lower = element[1]

        higher = lower + (current\_range \* char\_upper)

        lower = lower + (current\_range \* char\_lower)

    return lower

def decompress(code, intervals, length):

    decompressed = ''

    for \_ in range(length):

        print('Код:', code)

        for element in intervals:

            if code >= element[1] and code < element[2]:

                decompressed += element[0]

                print('Буква:', element[0])

                char\_upper = element[2]

                char\_lower = element[1]

                current\_range = char\_upper - char\_lower

                code = (code - char\_lower)/current\_range

                break

    return decompressed

Листинг 1 – Задание 2

**Задание 3:** Дать оценку возможности переполнения при выполнении вычислений. Для выполнения этого задания код, разработанный в предыдущем задании был дополнен следующим кодом:

def get\_overflow\_pos(seq, decompressed):

    pos = 0

    for i in range(len(seq)):

            if seq[i] != decompressed[i]:

                pos = i

                print('Позиция переполнения:', pos)

                print('Исходный символ:',seq[i])

                print('Полученный:',decompressed[i])

                break

Листинг 2 – Задание 3

**Задание 4:** Сравнить характеристики арифметического сжатия с вероятностными алгоритмами. Для выполнения этого задания код, разработанный в предыдущем задании был дополнен следующим кодом:

 efficiency = (len(str(compressed)) + len(seq))/ len(seq)

 print('\nЭффективность сжатия арифметическим методом:', efficiency)

 task\_shannon\_fano(seq, probs)

Листинг 3 – Задание 4

Результат работы итогового кода:

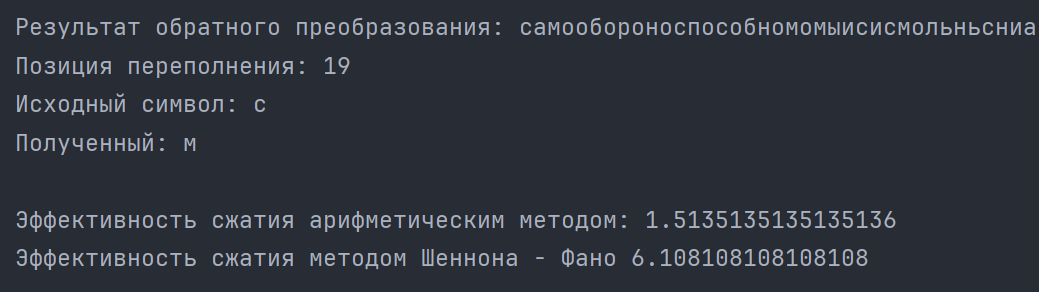
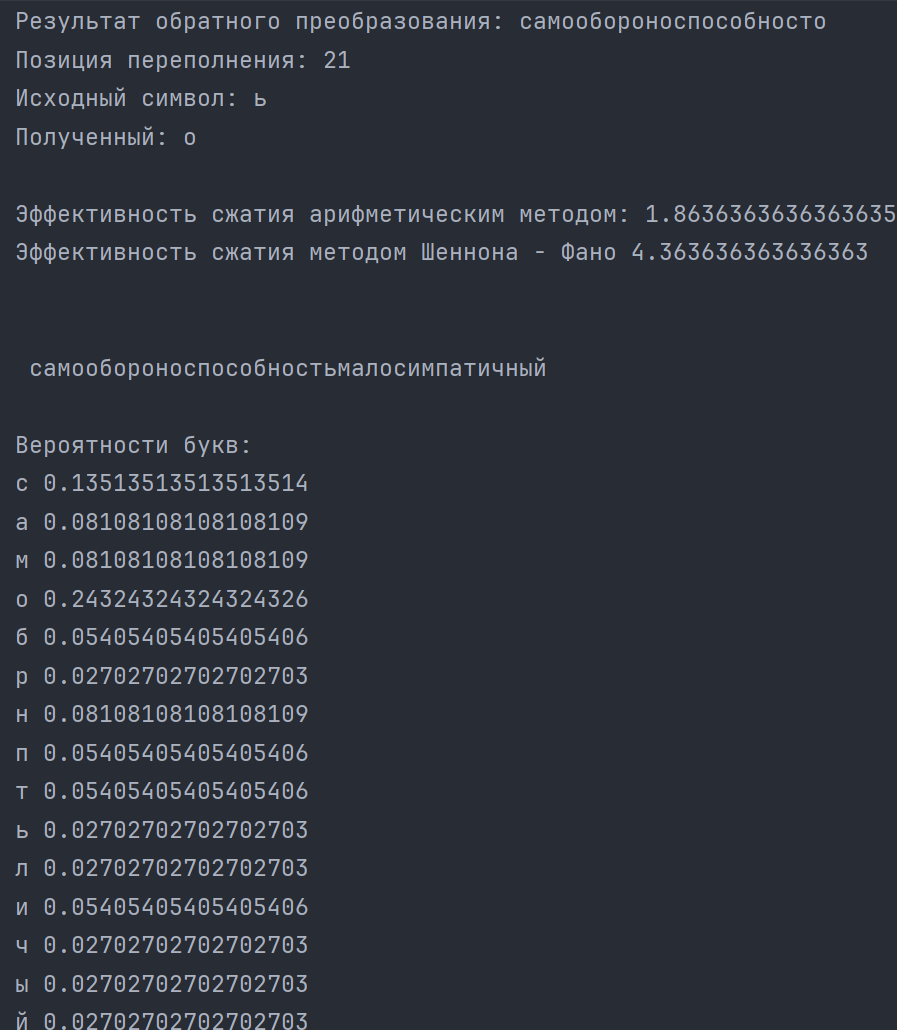
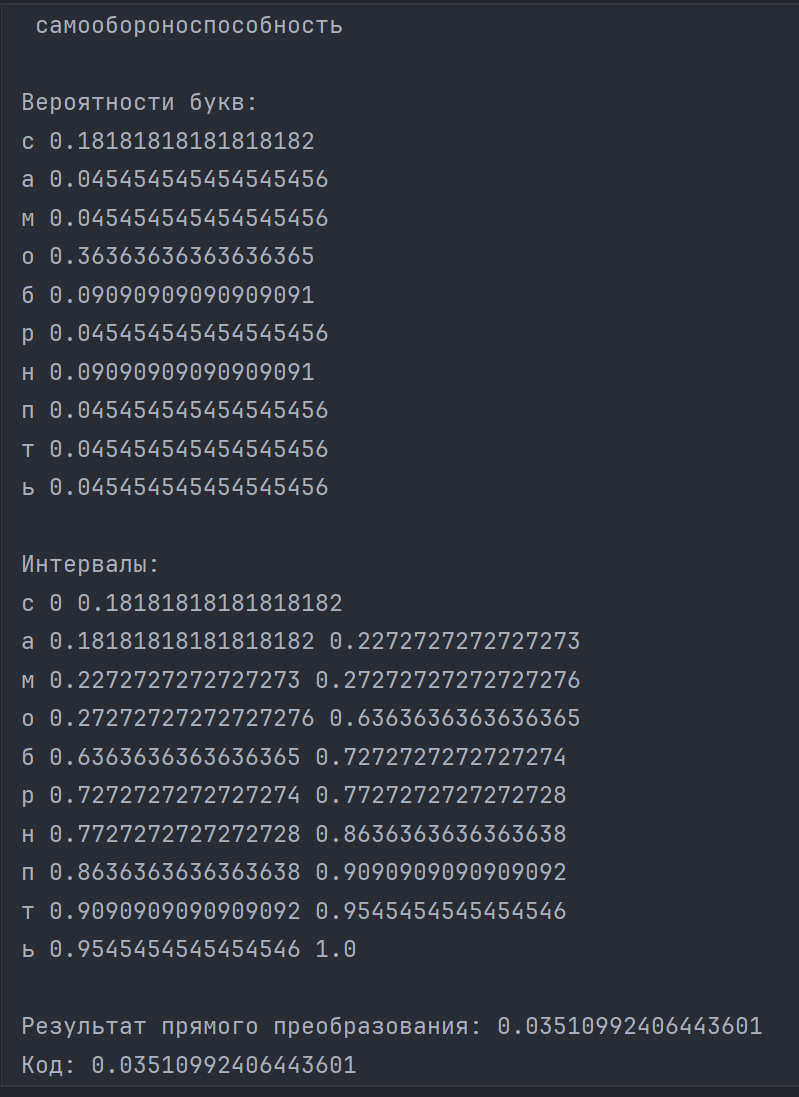


Рисунок 1 ­­­­– Результат работы программы

**Вывод:** в ходе лабораторной работы были приобретены практические навыки использования арифметических методов. Было выяснено, что в длинных сообщениях может возникнуть переполнение из-за вычисления значений в десятичных дробях. Эффективность арифметического сжатия ниже эффективности вероятностных алгоритмов.