Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский

политехнический университет»

(ПНИПУ)

Кафедра вычислительной математики и механики

Отчёт

по дисциплине: «Технологии обработки информации»

по теме: «Шифр Шеннона»

Выполнил:

Студент группы ИСТ-19-1бзу

Соболь Евгений Валерьевич

Проверил:

доцент кафедры ВМиМ

Сабатулина Татьяна Леонидовна

Пермь

2021 г.

**Листинг программы с комментариями по каждой функции:**

from math import copysign, fabs, floor, isfinite, modf, log, ceil  
from collections import Counter  
  
  
def float\_to\_bin(f: float) -> str:  
 *"""Функция преобразования вещественного числа из 10 с.ч в 2 с.ч"""* if not isfinite(f):  
 return repr(f) # inf nan  
 sign = '-' \* (copysign(1.0, f) < 0)  
 frac, fint = modf(fabs(f)) # split on fractional, integer parts  
 n, d = frac.as\_integer\_ratio() # frac = numerator / denominator  
 assert d & (d - 1) == 0 # Модульный тест  
 return f'{sign}{floor(fint):b}.{n:0{d.bit\_length() - 1}b}'  
  
  
def sorted\_dict(dict\_original: dict) -> dict:  
 *"""Функция сортировки словаря по значению"""* sort\_dict = {}  
 sort\_values = sorted(dict\_original.values(), reverse=True)  
 for i in sort\_values:  
 for j in dict\_original.keys():  
 if dict\_original[j] == i:  
 sort\_dict[j] = dict\_original[j]  
 return sort\_dict  
  
  
def lx(Px: float) -> int:  
 *"""Вычисляем Lx"""* return ceil(-log(Px, 2))  
  
  
def kod(bx: str, lx: int) -> str:  
 *"""Вычисляем код для каждого символа"""* if lx == 0:  
 return '0'  
 if bx.split(".")[1] == '0':  
 return '0' \* lx  
 return bx.split(".")[1][:lx]  
  
  
def beautiful\_output(d: dict, step: int):  
 *"""Функция для вывода результатов, 1-3, 5 шагов"""* if step == 1:  
 print("-----ПОДСЧЁТ СИМВОЛОВ-----")  
 for i in d:  
 print(f"Символ {i} встречается {d[i]}")  
 print()  
 if step == 2:  
 print("-----ПОДСЧЁТ ЧАСТОТЫ ПОЯВЛЕНИЯ-----")  
 for i in d:  
 print(f"Символ {i} c частотой {d[i]}")  
 print()  
 if step == 3:  
 print("-----СОРТИРОВКА ПО УБЫВАНИЮ-----")  
 k = 1  
 for i in d:  
 print(f"{k} 'элемент - символ {i} c частотой {d[i]}")  
 k += 1  
 print()  
 if step == 5:  
 print("-----СПИСКИ С ГОТОВЫМИ ЗНАЧЕНИЯМИ ДЛЯ ПОСТРОЕНИЯ КОДА-----")  
 for i in d:  
 print(f"Для символа {i} список {d[i]}")  
 print()  
  
  
def generic\_dict(line: str) -> dict:  
 *"""Создаём словарь со всеми основными вычислениями по каждому символу"""* final\_dict = dict(Counter(line))  
 beautiful\_output(final\_dict, 1)  
 for i in final\_dict.keys():  
 final\_dict[i] = final\_dict[i] / len(line)  
 beautiful\_output(final\_dict, 2)  
 final\_dict = sorted\_dict(final\_dict)  
 beautiful\_output(final\_dict, 3)  
 pre\_sum = [0]  
 j = 0  
 for i in final\_dict.values():  
 j += i  
 pre\_sum.append(j)  
 print("-----СПИСОК С ИНТЕРВАЛАМИ-----")  
 print(pre\_sum, "\n")  
 j = 0  
 k = 0  
 for i in final\_dict.keys():  
 final\_dict[i] = [final\_dict[i]]  
 final\_dict[i].append(pre\_sum[k])  
 final\_dict[i].append(float\_to\_bin(final\_dict[i][-1]))  
 final\_dict[i].append(lx(final\_dict[i][0]))  
 final\_dict[i].append(kod(final\_dict[i][2], final\_dict[i][-1]))  
 j += 1  
 k += 1  
 beautiful\_output(final\_dict, 5)  
 return final\_dict  
  
  
def generic\_kod(line: str, dictionary: dict) -> str:  
 *"""Преобразуем строку в шифр по словарю"""* output\_dict(dictionary)  
 result = ""  
 for i in line:  
 result += dictionary[i][-1]  
 print("--------------------РЕЗУЛЬТАТ--------------------")  
 return result  
  
  
def shannon\_code(line: str) -> str:  
 *"""Преобразуем строку в код методом Шеннона используя созданный словарь"""* abc = generic\_dict(line)  
 cipher = generic\_kod(line, abc)  
 return cipher  
  
  
def output\_dict(d: dict):  
 *"""Функция отображения кодов символов"""* print("--------------------КОДЫ СИМВОЛОВ--------------------")  
 for i in d.keys():  
 print(f'Символ "{i}" - "{d[i][-1]}"')  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 print("Задание № 1 Шифр Шеннона")  
 source\_line = input("Введите строку: ")  
 print(shannon\_code(source\_line))

**Пример работы программы № 1:**

Тестовое значение для ввода: Hello world!1

**Результат:**

01100111000000001100010010011011000110011011110

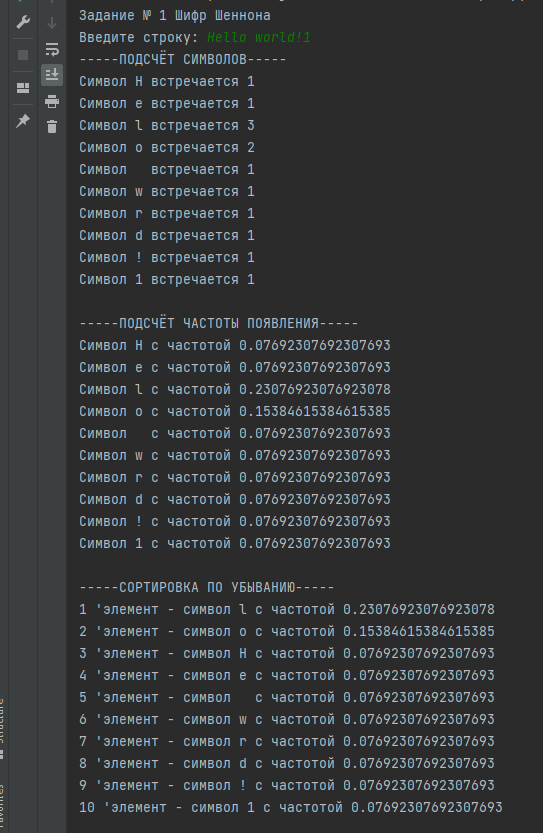
****

Рисунок **1** - Скриншот результатов (1)

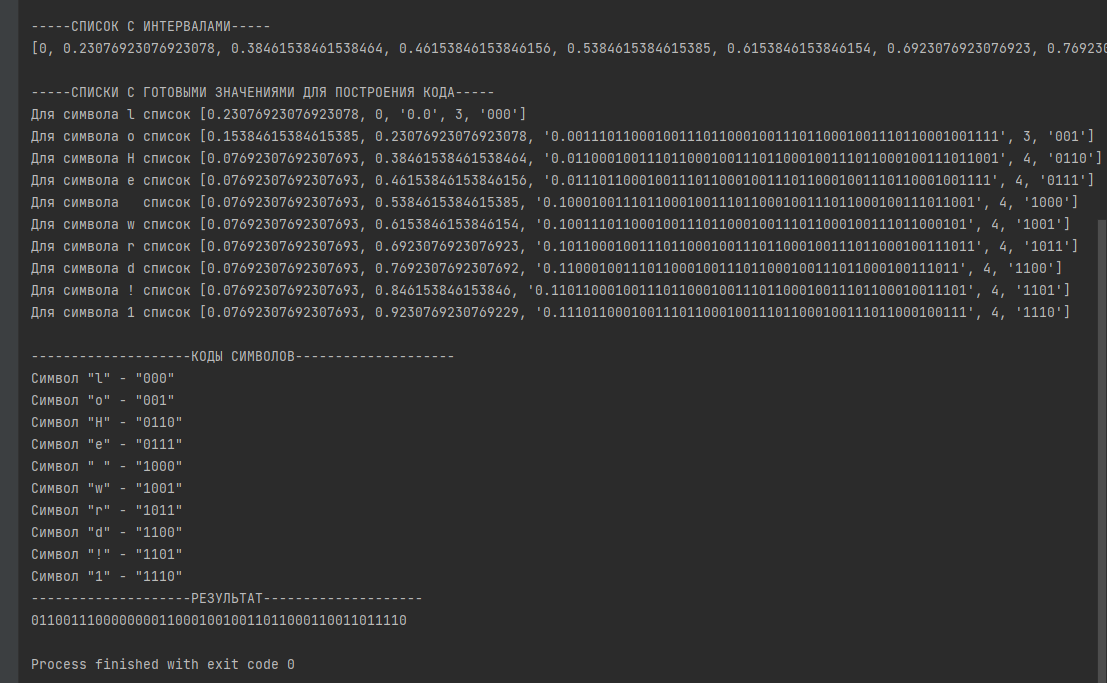


Рисунок 2 - Скриншот результатов (2)

**Пример работы программы № 2:**

**Пример 2.6.1 из учебника Кудряшова Б.Д стр. 71**

Тестовое значение для ввода: aaaaaaaaaaaaaabbbbbbbbccccccddddeeeeffff

**Результат:**

0000000000000000000000000000010010010010010010010010100100100100100100101110111011101111001100110011001110111011101110

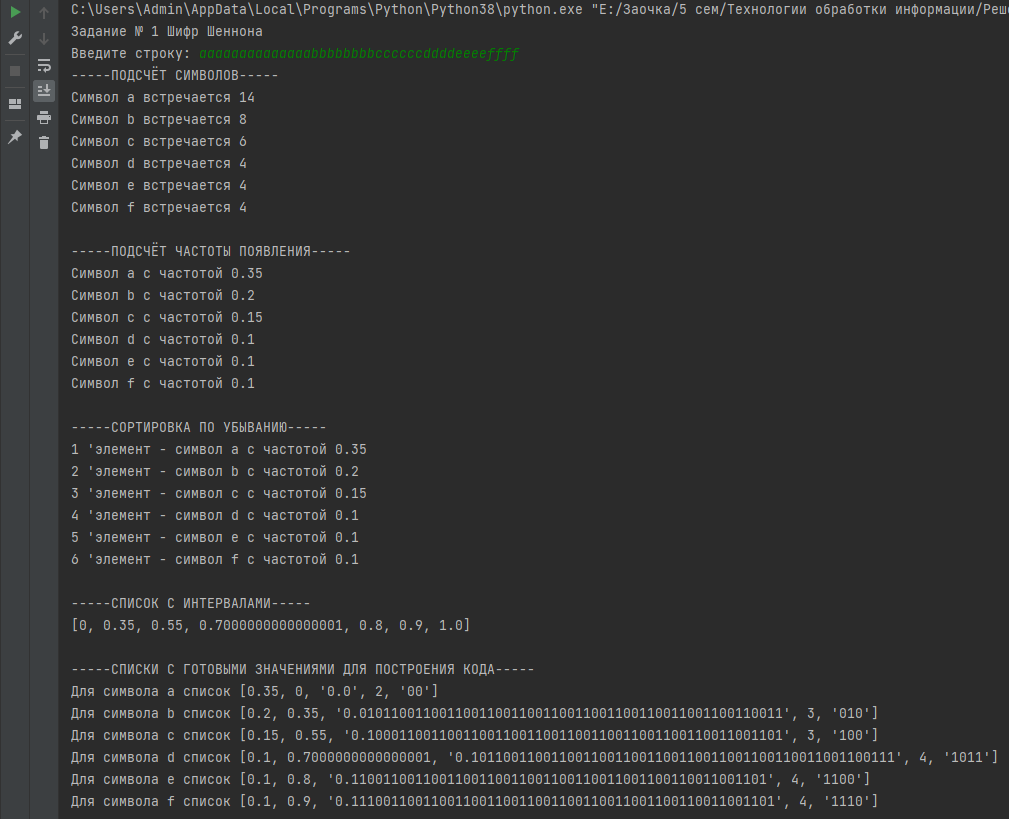


Рисунок 3 – Скриншот результатов (3)

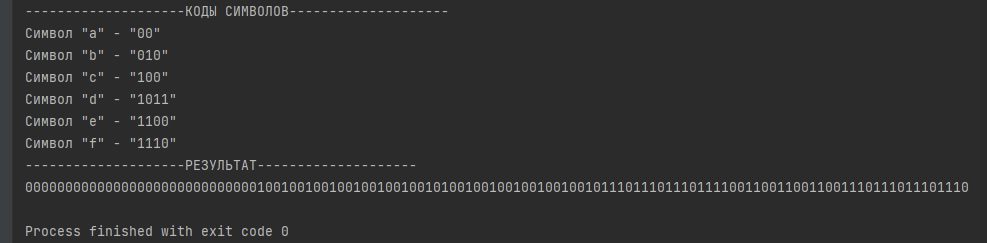


Рисунок - Скриншот результатов (4)