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 №7 НА ТЕМУ:**

**Перемежение/деперемежение данных в информационно-вычислительных системах**

Выполнила студентка 3 курса 1 группы

Пригодич Вера Валерьевна

Минск 2022

**Задание 1:** Необходимо разработать авторское приложение в соответствии с целью лабораторной работы. По умолчанию используется блочный перемежитель/деперемежитель. По желанию студент может использовать иной. Задание выполняется по указанию преподавателя в соответствии с вариантом из таблицы.

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

# формирует матрицу информационных слов заданной длины

def div\_inf\_words(seq, length):

   # Добавляем нули слева

   if len(seq) % length != 0:

      for \_ in range(length - len(seq) % length):

         seq = '0' + seq

   inf\_words = [''.join(i) for i in grouper(seq, length)]

   return inf\_words

# формирует матрицу перемежения

def generate\_interleave\_matrix(seq: str, col\_num: int):

   # Добавляем нули слева

   if len(seq) % col\_num != 0:

      for \_ in range(col\_num- len(seq) % col\_num):

         seq = '0' + seq

   matrix = list(seq)

   row\_num = int(len(seq)/col\_num)

   return np.reshape(matrix, (row\_num, col\_num))

# перемежение

def interleave(matrix):

   return ''.join(map(str, matrix.transpose().ravel()))

# деперемежение

def deinterleave(y: str, col\_num: int) -> str:

    matrix = []

    for \_ in range(int(len(y) / col\_num)):

        matrix.append('')

    # Формируем матрицу по столбцам

    for \_ in range(col\_num):

        for i in range(len(matrix)):

            matrix[i] += y[0]

            y = y[1:]

    return ''.join(matrix)

Листинг 1 – задание 1

**Задание 2:** Местоположение заданной группы ошибок выбирается (генерируется) случайным образом. Необходимо для группы ошибок каждой длины сгенерировать 30−40 случайных ситуаций. После деперемежения и исправления ошибок в сообщении сравнить передаваемую последовательность и полученную после исправления ошибок. Проанализировать эффективность перемежения/деперемежения. Для выполнения этого задания код, разработанный в предыдущем задании был дополнен следующим кодом:

# формирует случаный пакет ошибок заданной длины

def rand\_mistakes(interleaved, num):

   y = [int(x) for x in interleaved]

   place = random.randint(0, len(interleaved) - num - 1)

   for i in range(place, place + num):

      if (y[i] == 0):

         y[i] = 1

      else:

         y[i] = 0

   y = ''.join(map(str, y))

   return y

# исправление ошибок

def correct\_mistakes(seq, g, check\_matrix, n):

   inf\_words = [''.join(i) for i in grouper(seq, n)]

   corrected\_words = []

   for word in inf\_words:

      syndrome = count\_syndrome(word, g)

      error\_vector = error\_position(syndrome, check\_matrix, n)

      corr\_word = xor(word, error\_vector)

      corrected\_words.append(corr\_word)

   return ''.join(corrected\_words)

print("Корректирующие способности:")

# Для 3 ошибок

total = 100

right = 0

for \_ in range(total):

    mistaken = rand\_mistake(interleaved, 3)

    deinterleaved = deinterleave(mistaken, col\_num)

    corrected = correct\_mistakes(deinterleaved[number\_of\_added\_zeros:], g, check\_matrix, n)

    if (corrected == encoded):

        right += 1

print(f"Для 3 ошибок: {right / total}")

Листинг 2 – задание 2

Результат работы итогового кода:

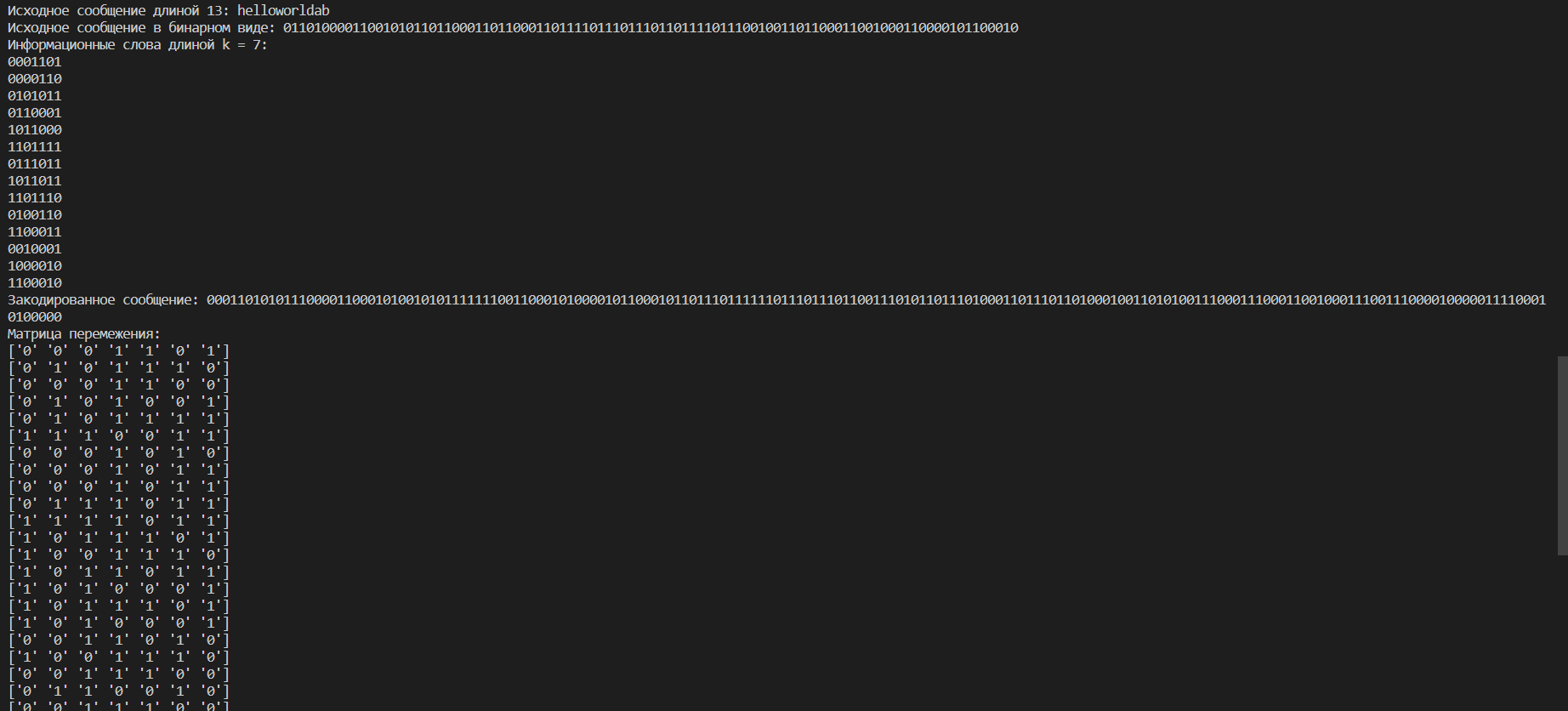


Рисунок 1 ­­­­– Результат работы программы

Анализ корректирующей способности:

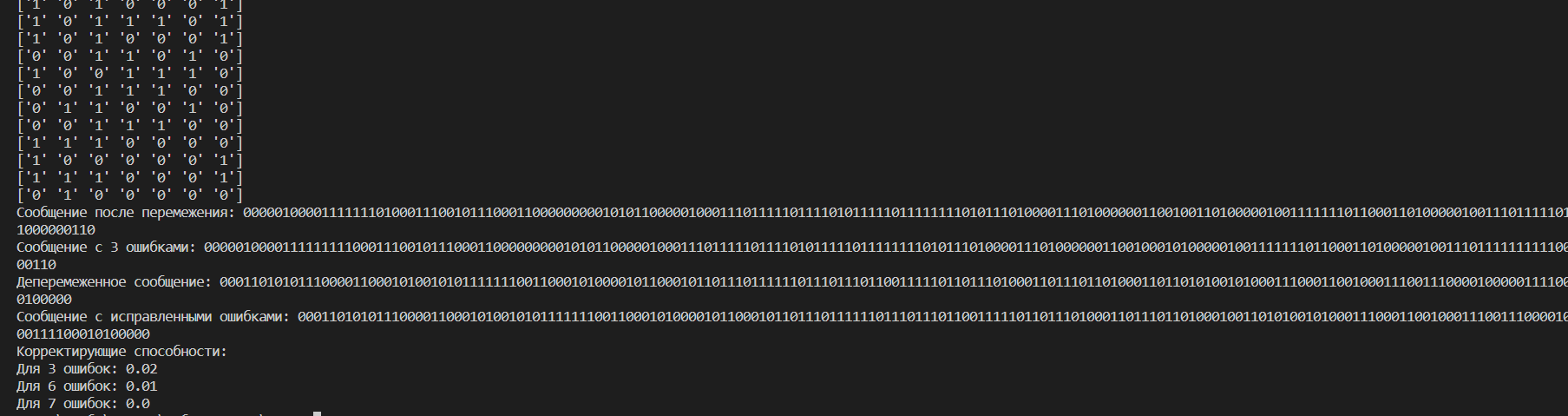


Рисунок 2 ­­­­– Корректирующая способность кода

С увеличением длины пакета ошибок снижается эффективность перемежения/деперемежения.

**Вывод:** в ходе лабораторной работы были приобретены практические навыки использования методов перемежения/деперемежения двоичных данных в информационных системах.