Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №1**

Дисциплина: «Разработка профессиональных приложений»

Обработка списков

Вариант № 10

Выполнила:

студентка группы ИВТАСбд-21

Микка Е. И.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту №10.**

Работа с двумя списками А и В, создаваться они могут как пользователем (ввод с клавиатуры), так и путём автоматической генерации. Между списками необходимо поменять самые длинные цепочки из четных элементов. Самая длинная цепочка из четных элементов списка А встаёт на место самой длинной цепочки из четных элементов списка В и наоборот. По мимо этого к программе есть следующие требования: некорректный ввод должен корректно обрабатываться программой, код должен быть откомментирован, реализация правильной декомпозиции программы на методы и необходимо реализовать заданный алгоритм двумя способами: (методами) без стандартных функций и со стандартными функциями.

**Описание реализации.**

В независимости от того какой способ создания списков выбрал пользователь, по итогу ввода исходных данных мы получаем две строки. Одна будет далее преобразовано в список А, а другая – в список B. В случае ввода данных с консоли просто считываются строки. Если же была выбрана автоматическая генерация, то сначала генерируется рандомная длина списка, а после создаётся соответствующие число элементов, они записываются в строку через пробел. Полученные строки сначала проверяются на пустоту, если в них информация или же с консоли «списалась» пустая строка. Далее осуществляется проверка на то, чем является имеющиеся введенные элементы (данная проверка осуществляется только в случае, если первая проверка прошла успешно). Если и эта проверка прошла успешно, то строки спокойно конвертируются в список, состоящий из целочисленных переменных.

Сам алгоритм как было выше указано написать двумя методами без стандартных функций и со стандартными функциями. Оба метода работают по идентичной схеме, с той лишь разницей, что в алгоритме без стандартных функций эти самые функции описаны самостоятельно, в программе.

Выполнение алгоритма:

Для каждого списка создаётся дополнительный список, а также дополнительная переменная. Отдельно рассматривается первый элемент основного списка. Если этот элемент чётный, то в дополнительный список добавляется 1, а в дополнительную переменную номер этого элемента в списке. Далее с помощью цикла for происходит проход по всему списку следующим образом:

Если i-ый элемент – четный, то в дополнительный список добавляется значение предыдущего элемента, увеличенное на единицу, а в дополнительную переменную записывается номер элемента в списке. Если же i-ый номер – нечетный, то в дополнительный список записывается ноль.

Таким образом, найдя в дополнительном списке максимальный элемент, мы найдем максимальную длину четной последовательности, а также нам дан номер элемента в списке, на котором эта последовательность заканчивается. Всё что остается это создать два списка, скопировав в один список A, а в другой список B, и поменять «взять» из первоначальных списков нужные нам фрагменты и «вставить» их в только что созданные списке. Остается лишь вывести получившиеся списки.

**Описание возникших затруднений.**

Основная трудность заключалось в том, как записать необходимую информацию о последовательности и как её «ввести» в список, какие имеющиеся возможности языка для этого использовать. Создание списка, создание функций затруднений не вызвали.

**Описание альтернативных способов решения.**

Можно было, создать два пустых списка. Циклом for пройтись по исходным спискам и найти для каждого списка номер элемента в списке, с которого начинается последовательность, и номер элемента в списке, на котором последовательность заканчивается.

Второй раз пройтись циклом, предположим, по пустому списку C, который будет служит переформированным списком A, и переписывать в него элементы списка A до тех пор пока цикл не дойдёт до элементов найденным прошлым проходом по списку. В таком случае, цикл переходит на выполнение вложенного цикла, и уже идёт переписывание элементов не из списка A, а из списка B.

Пример:

Список A:

3 1 0 4 2 3 8

Список B:

4 8 6 0 7 7 9

После первого цикла:

a\_min = 2 a\_max = 4

b\_min = 0 b\_max = 3

Второй цикл (Список C – пуст):

I = 2 C = 3 1

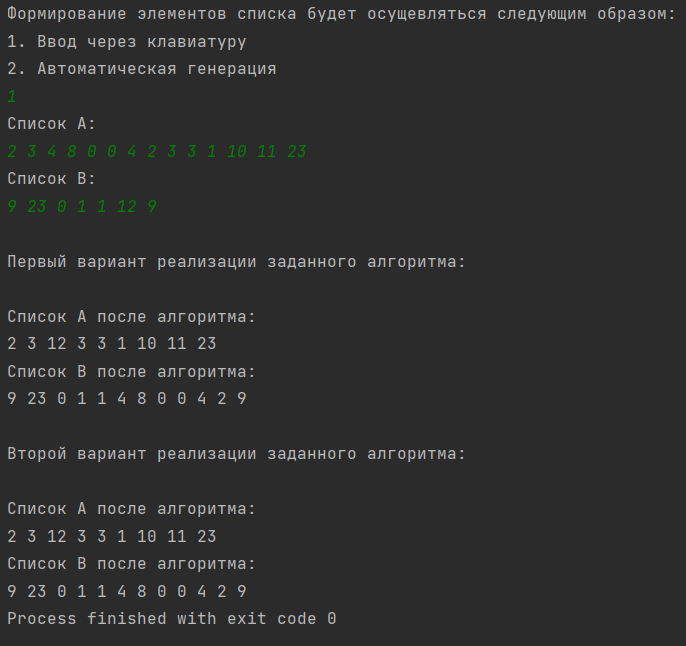
Теперь проходим уже по списку B с элемента 0 до элемента 3.

C = 3 1 4 8 6 0

Далее I присваиваем значение a\_max и продолжаем переписывать список А до конца. Работа со списком B аналогична.

**Пример работы программы.**

Ввод исходных данных осуществляется путём ввода с клавиатуры:



Ввод исходных данных осуществляется путём автоматической генерации:

