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 19

Выполнил:

студент группы ИВТАПбд-21

Романов Н.Ю

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

Вариант 19:

1. Из списка удалить элементы, встречающиеся в списке только один раз.

Пример: из списка A[7]: 9 1 4 9 1 9 0 должен получиться список A[5]: 9 1 9 1 9.

1. Ввод элементов списка должен быть доступен с клавиатуры и путем автоматической генерации
2. Программа должна корректно обрабатывать некорректный ввод.
3. Исходный код должен быть откомментирован
4. Необходимо реализовать правильную декомпозицию программы на методы.
5. Необходимо реализовать предложенный алгоритм без использования стандартных функций и с использованием (необходимо реализовать два отдельных метода).

**Описание реализации**

Для ввода массива была создана функция input\_list() и remove\_unique\_elements(lst), в input\_list() использовалась встроенная функция input() с преобразованием в тип данных int, а также были использованы списки для хранения данных массива. Программа обрабатывает ввод благодаря конструкции try-except, в котором при появлении исключения ValueError выводится в консоль сообщение о некорректности ввода, после чего программа завершается.

Листинг 1. Функция input\_list()

|  |
| --- |
| def input\_list():  """  Функция запрашивает у пользователя количество элементов в списке и сами элементы.  Возвращает список чисел.  """  while True:  try:  n = int(input("Введите количество элементов в списке: "))  lst = []  for i in range(n):  elem = int(input(f"Введите {i + 1}-й элемент списка: "))  lst.append(elem)  return lst  except ValueError:  print("Некорректный ввод. Попробуйте еще раз.") |

В функции myhandinput() ввод элементов массива осуществляется через пробел, то есть вводится строка, которая в последствии с созданной функцией mysplit(array) разделяет введённые числа по пробелам на список.

Листинг 2. Функция remove\_unique\_elements(lst)

|  |
| --- |
| def remove\_unique\_elements(lst):  """  Функция удаляет из списка элементы, встречающиеся только один раз.  Возвращает измененный список.  """  # создаем словарь, где ключ - элемент списка, значение - количество его вхождений в список  count\_dict = {}  for elem in lst:  if elem in count\_dict:  count\_dict[elem] += 1  else:  count\_dict[elem] = 1    # создаем новый список, содержащий только те элементы, которые встречаются несколько раз  new\_lst = [elem for elem in lst if count\_dict[elem] > 1]    return new\_lst |

Для реализации основного алгоритма программы был использован цикл, созданная функция поиска минимального элемента а также метод .append(), .remove() и .clear(). Алгоритм работает следующим образом:

1. После ввода массива циклом перебираются все элементы списка и записываются в третий до тех пор, пока не найдётся чётное число.
2. Если находится чётное число, то оно записывается во второй список для последовательности чётных чисел, пока не найдётся нечётное число
3. После нахождения нечётного числа находится минимальный элемент в последовательности, после чего удаляется методом .remove(), который удаляет первый попавшийся искомый элемент в списке.
4. Получившаяся обработанная последовательность записывается в третий список, в котором будет хранится конечный массив.
5. Если осталась ещё не обработанная часть массива, то цикл повторяется.

**Описание возникших затруднений**

Трудности возникли с переобучением стандартам синтаксиса языков C/C++ и Java, в которых использование точки с запятой, фигурных скобок, последовательность объявления функций и переменных обязательна. А также трудности возникли с привыканием к тому что язык Python динамичный, то есть не требуется указывать тип данных переменных, а также есть простая возможность в их преобразовании.

**Описание альтернативных способов решения**

Данный алгоритм можно было реализовать иначе, без использования дополнительных списков. Можно было сохранять индекс начала и индекс конца последовательности чётных чисел, и уже по данному диапазону удалять минимальный элемент. А ввод можно было осуществить по посимвольному вводу элементов.

Тестирование

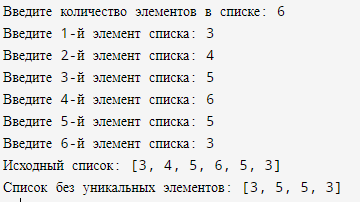


Рис 1. Ручной ввод массива

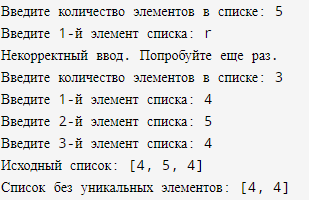


Рис 2. Обработка некорректного ввода данных

Вывод

В данной лабораторной работе были приобретены базовые навыки программирования на языке Python, в частности в работе со списками и преобразованием типов данных. В результате была сделана программа, которая из введённого списка пользователем удаляла минимальный элемент из каждой подпоследовательности чётных чисел.

**Код программы**

from random import randint

def input\_list():

"""

Функция запрашивает у пользователя количество элементов в списке и сами элементы.

Возвращает список чисел.

"""

while True:

try:

n = int(input("Введите количество элементов в списке: "))

lst = []

for i in range(n):

elem = int(input(f"Введите {i + 1}-й элемент списка: "))

lst.append(elem)

return lst

except ValueError:

print("Некорректный ввод. Попробуйте еще раз.")

def remove\_unique\_elements(lst):

"""

Функция удаляет из списка элементы, встречающиеся только один раз.

Возвращает измененный список.

"""

# создаем словарь, где ключ - элемент списка, значение - количество его вхождений в список

count\_dict = {}

for elem in lst:

if elem in count\_dict:

count\_dict[elem] += 1

else:

count\_dict[elem] = 1

# создаем новый список, содержащий только те элементы, которые встречаются несколько раз

new\_lst = [elem for elem in lst if count\_dict[elem] > 1]

return new\_lst

if \_\_name\_\_ == "\_\_main\_\_":

# получаем список от пользователя

my\_list = input\_list()

print(f"Исходный список: {my\_list}")

# удаляем уникальные элементы

new\_list = remove\_unique\_elements(my\_list)

print(f"Список без уникальных элементов: {new\_list}")