

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра информационных систем

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4
Алгоритмы обработки списков

Вариант 2

Преподаватель

_____ А.Г. Фельдман
подпись, дата

Студент

КИ22-06Б, 032215878
номер группы, зачетной книжки

_____ Д.А. Безпалый
подпись, дата

Красноярск 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ОСНОВНАЯ ЧАСТЬ	4
Задание 1	4
Задание 2	5
Задание 3	6
Задание 4	7
ЗАКЛЮЧЕНИЕ.....	8
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	9

ВВЕДЕНИЕ

Цель практической работы: изучение алгоритмов обработки списков в Python.

Основная задача практической работы: решить 3 задачи и 1 задачу по своему варианту на платформе «Информатикс».

ОСНОВНАЯ ЧАСТЬ

Задание 1

Удаление экстремальных значений.

При анализе собранных по результатам научных экспериментов данных зачастую возникает необходимость избавиться от экстремальных значений, прежде чем продолжать двигаться дальше. Напишите функцию, создающую копию списка с исключенными из него n наибольшими и наименьшими значениями и возвращающую ее в качестве результата. Порядок следования элементов в измененном списке необязательно должен в точности совпадать с источником.

Код программы изображен на рисунке 1.

```
def remove_extremes(data, n):  
    if n < 1 or len(data) <= 2 * n:  
        return data.copy()  
  
    sorted_data = sorted(data)  
    return sorted_data[n: -n]  
  
original_list = [1, 2, 3, 4, 5, 100, -50]  
n = 1  
new_list = remove_extremes(original_list, n)  
print(new_list)
```

Рисунок 1 – Программный код

Задание 2

Удаление дубликатов.

Разработайте программу, в которой у пользователя будет запрошен список слов, пока он не оставит строку ввода пустой. После этого на экране должны быть показаны слова, введенные пользователем, но без повторов, – каждое по одному разу. При этом слова должны быть отображены в том же порядке, в каком их вводили с клавиатуры.

Код программы изображен на рисунке 2.

```
def unique_words():
    words = []
    seen = set()

    while True:
        word = input("Введите слово (или нажмите Enter, чтобы закончить): ")
        if word == "":
            break
        if word not in seen:
            seen.add(word)
            words.append(word)

    print("Уникальные слова:")
    for word in words:
        print(word)

unique_words()
```

Рисунок 2 – Программный код

Задание 3

Двоичный поиск.

В первой строке входных данных содержатся натуральные числа N и K ($0 < N, K \leq 100000$). Во второй строке задаются N элементов первого массива, отсортированного по возрастанию, а в третьей строке – K элементов второго массива. Элементы обоих массивов - целые числа, каждое из которых по модулю не превосходит 10^9 . Требуется для каждого из K чисел вывести в отдельную строку "YES", если это число встречается в первом массиве, и "NO" в противном случае.

ID	Участник	Задача	Дата	Язык	Статус	Пройдено тестов	Баллы	Подробнее
37571679	Денис Безпальный	4. Двоичный поиск	2024-10-10 17:32:03	Python 3.9	<input type="text" value="OK"/>	8	100	Подробнее

Рисунок 3 – Ход решения задачи

Код, прошедший проверку изображен на рисунке 4.

```
1 import bisect
2
3 def binary_search_in_sorted_array():
4     N, K = map(int, input().split())
5
6     sorted_array = list(map(int, input().split()))
7
8     search_elements = list(map(int, input().split()))
9
10    result = []
11    for element in search_elements:
12        index = bisect.bisect_left(sorted_array, element)
13
14        if index < N and sorted_array[index] == element:
15            result.append("YES")
16        else:
17            result.append("NO")
18
19    print("\n".join(result))
20
21 binary_search_in_sorted_array()
```

Рисунок 4 – Программный код

Задание 4

Задача 112293.

Условия задачи: Напишите программу, которая находит в массиве минимальный и максимальный из чётных положительных элементов.

Ход решения задачи изображен на рисунке 5.

ID	Участник	Задача	Дата	Язык	Статус	Пройдено тестов	Баллы	Подробнее
37571552	Денис Безпалый	112293. Максимум и минимум	2024-10-10 17:22:27	Python 3.9	<input type="text" value="OK"/>	23	100	Подробнее

Рисунок 5 – Ход решения задачи

Код, прошедший проверку изображен на рисунке 6.

```
1  def find_min_max_even_positive():
2      N = int(input())
3
4      elements = list(map(int, input().split()))
5
6      min_even = float('inf')
7      max_even = float('-inf')
8
9      for num in elements:
10         if num > 0 and num % 2 == 0:
11             if num < min_even:
12                 min_even = num
13             if num > max_even:
14                 max_even = num
15
16         if min_even == float('inf') or max_even == float('-inf'):
17             print(-1, -1)
18         else:
19             print(min_even, max_even)
20
21  find_min_max_even_positive()
```

Рисунок 6 – Программный код

ЗАКЛЮЧЕНИЕ

Выполняя практическую работу, были получены навыки работы с алгоритмами обработки списков на языке программирования Python.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СТУ 7.5–07–2021. Стандарт университета «Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности».
2. eКурсы – Система электронного обучения СФУ : Курс: Основы программирования на Python 2024 URL: <https://e.sfu-kras.ru/course/view.php?id=38620#section-0> (дата обращения 01.10.2024).