

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ
«МИФИ»

Отчет по лабораторной работе №1
по курсу: «Специальные технологии баз данных»

Выполнил: студент группы С20-702

Нуриддинходжаева А.А.

(подпись)

(Фамилия И.О.)

Проверил:

Манаенкова Т.А.

(оценка)

(подпись)

(Фамилия И.О.)

Москва 2023 г

Условия заданий

Вариант 4

Задача 1

Напишите программу, определяющую является ли введённое целое число идеальным по какому-либо из следующих трёх определений: - число идеально, если сумма его делителей при разложении на простые делители, отличных от него самого и единицы равна самому числу ($4 = 2 * 2 = 2 + 2$); - число идеально, если сумма его делителей при разложении на простые делители, отличных от него самого и единицы равна сумме его цифр ($27 \Rightarrow$ сумма цифр $2 + 7 = 9$, $27 = 3 * 3 * 3 \Rightarrow$ сумма делителей $= 9$). - в соответствии с понятием совершенного числа: https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B2%D0%B5%D1%80%D1%88%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE.

Задача 2

Напишите программу представления чётного числа (большого или равного 8) в виде суммы четырёх простых чисел.

Задача 3

Реализуйте на языке Python программу, осуществляющую кодирование текста на основе шифра Виженера. Вход: 2 строки – текст и ключ (последовательность английских букв). Выход – зашифрованный текст. Ключ может быть короче или длиннее текста. Также реализуйте аналогичную программу для дешифрования.

Задача 4

Напишите программу, которая находит во введённой строке английских символов самую длинную последовательность символов, идущих в алфавитном порядке. Например, для “abcdlhabkefghijklty” самая длинная последовательность efg hijkl.

Решение

Задача 1

```
import sympy as sp
import numpy as np

def sum_d(n):
    dels = []
    i = 2
    while n != 1:
        if n % i == 0 and sp.isprime(i):
            dels.append(i)
            n //= i
        else:
            i += 1
    return np.sum(dels)

def sum_c(n):
    sum = 0
    while n > 0:
        d = n % 10
        sum += d
        n //= 10
    return sum

x = int(input("введите число: "))

if sum_d(x) == x:
    print("Число идеальное по первому определению")
elif sum_c(x) == sum_d(x):
    print("Число идеальное по второму определению")
elif sp.is_perfect(x):
    print("Число идеальное по третьему определению")
else:
    print("Число неидеальное")
```

Задача 2

```
def check(number): #Функция проверяет, что число больше 8
    if (number > 8 and number%2 == 0):
        print('Число принято\n')
        return number
    else:
        print('Попробуй ввести число снова\n')
        number = int(input())
        number = check(number)
    return number

def is_prime(en): #Функция проверяет, является ли число простым
    if en < 2:
        return False
    for i in range(2, int(en ** 0.5) + 1):
        if en % i == 0:
            return False
    return True

n = int(input("Введите число: "))
n = check(n)
```

```

for i in range(2, n - 7):
    if is_prime(i):
        # Если первое число простое, перебираем второе число от i+1 до n-5
        for j in range(i + 1, n - 5):
            if is_prime(j):
                # Если второе число простое, перебираем третье число от j+1 до n-2
                for k in range(j + 1, n - 2):
                    if is_prime(k):
                        # Если третье число простое, проверяем сумму первых трех чисел и ищем четвертое число
                        if i + j + k < n and is_prime(n - i - j - k):
                            print(f"{n} = {i} + {j} + {k} + {n - i - j - k}")
exit() # выходим из программы после первого найденного решения

```

Задача 3

```

f = int(input("Choose option: \n 1 - encryption \n 2 - decryption \n"))
alph = "abcdefghijklmnopqrstuvwxyz"

if f == 1:
    phr = str(input("Input phase: \n"))
    k = str(input("Input key: \n"))
    enc = ""
    if len(k) < len(phr):
        i = 0
        while len(k) < len(phr):
            k = k[:] + k[i]
            i += 1

    if len(k) > len(phr):
        k = k[1:len(phr)]
    j = 0
    while j < len(phr):
        enc = enc[:] + alph[(alph.index(phr[j]) + alph.index(k[j])) % len(alph)]
        j += 1
    print("Decrypted message:", enc)

else:
    enc = str(input("Input decripted message: \n"))
    k = str(input("Input key: \n"))
    phr = ""
    if len(k) < len(enc):
        i = 0
        while len(k) != len(enc):
            k = k[:] + k[i]
            i += 1

    if len(k) > len(enc):
        k = k[1:len(enc)]
    j = 0
    while j < len(enc):
        phr = phr[:] + alph[(alph.index(enc[j]) - alph.index(k[j]))]
        j += 1
    print("Encrypted message:", phr)

```

Задача 4

```

alphabet = "abcdefghijklmnopqrstuvwxyz"

```

```

inp_str = input("Enter a line in English: ")
c_str = inp_str[:]
inp_str = inp_str.lower()
current_longest = ""
longest = ""
for i in range(len(inp_str)):
    if inp_str[i] in alphabet:
        if len(current_longest) == 0:
            current_longest = inp_str[i]
        else:
            if alphabet.index(current_longest[-1]) + 1 == alphabet.index(inp_str[i]):
                current_longest += inp_str[i]
            else:
                if len(longest) <= len(current_longest):
                    longest = current_longest
                    current_longest = inp_str[i]
    else:
        if len(longest) <= len(current_longest):
            longest = current_longest
            current_longest = ""
if len(longest) <= len(current_longest):
    longest = current_longest
pos = inp_str.find(longest)
print('longest:', c_str[pos : len(longest) + pos])

```