**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П.О.СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

направление специальности 1-40 05 01-12 Информационные системы

и технологии (в игровой индустрии)

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине: « Скриптовые языки программирования»

на тему: «Введение в программирование на скриптовых языках»

Выполнил: студент гр. ИТИ-21

Важная Я.И.

Принял: преподаватель

Комракова Е.В.

Гомель 2024

Цель работы: научиться работать с условными операторами, использовать операторы циклов и итерируемые объекты при программировании различных практических задач на языке Python.

Задание:

1. Написать программы в соответствии с номером своего варианта.

**Ход работы**

Вводится натуральное число. Определить, является ли оно трехзначным. Если это так, то вывести «Трехзначное число», иначе – «Не трехзначное число». Программу реализовать без использования операторов циклов.

На рисунке 3 изображен результат выполнения 1-го задания (не трёхзначное ).



Рисунок 1 – Результат выполнения 1-го задания

На рисунке 4 изображен результат выполнения 1-го задания (трёхзначное).



Рисунок 2 – Результат выполнения 1-го задания

Вводятся два натуральных числа: a – время в секундах; b – время в минутах.

С помощью тернарного условного оператора определить наибольшее время из двух введенных и отобразить результат на экране.

На рисунке 3 изображен результат выполнения 2-го задания, когда минуты больше секунд.



Рисунок 3 – Результат выполнения 2-го задания

На рисунке 4 изображен результат выполнения 2-го задания, когда минуты равны секундам.



Рисунок 4 – Результат выполнения 2-го задания

На рисунке 5 изображен результат выполнения 2-го задания, когда секунды больше минут.



Рисунок 5 – Результат выполнения 2-го задания

Вводится список имен студентов в одну строчку через пробел. Определить, что хотя бы одно имя в этом списке начинается и заканчивается на ту же самую букву (без учета регистра). Реализовать программу с использованием операторов while и break. Вывести ДА, если условие выполняется, и НЕТ – в противном случае.

На рисунке 6 изображен результат выполнения 3-го задания (присутствует имя с одинаковыми буквами в начале и в конце).



Рисунок 6 – Результат выполнения 3-го задания

На рисунке 7 изображен результат выполнения 3-го задания (отсутствует имя с одинаковыми буквами в начале и в конце).



Рисунок 7 – Результат выполнения 3-го задания

В некоторой стране используются денежные купюры достоинством в 1,

2, 4, 8, 16, 32 и 64. Вводится натуральное число n. Как наименьшим количеством таких денежных купюр можно выплатить сумму n? Вывести

на экран список купюр для формирования суммы n. Предполагается, что имеется достаточно большое количество купюр всех достоинств.

На рисунке 8 изображен результат выполнения 3-го задания.



Рисунок 8 – Результат выполнения 4-го задания

На рисунке 8 изображен результат выполнения проверки на положительность 3-го задания.



Рисунок 9 – Результат выполнения 4-го задания

Вывод: в результате выполнения лабораторной работы научиться работать со строками и списками на языке Python, использовать генераторы списков, словари, кортежи и множества при программировании различных практических задач.

**ПРИЛОЖЕНИЕ А**

(обязательное)

Текст программы

**1ex.py:**

import math

x = int(input("Введите число: "))

x1 = 99

x2 = 1000

if x1 < x < x2:

print ("Трёхзначное")

else:

print ("Не трёхзначное")

**2ex.py:**

a = int(input("введите время в минутах: "))

b = int(input("Введите время в секундах "))

if (a\*60)>b:

print (f"{a} минут(ы) - больше")

elif (a\*60)<b:

print (f"{b} секунд - больше")

else:

print ("Равны")

**3ex.py:**

students = input("Names: ").split()

i = 0

while i < len(students):

name = students[i]

if name[0].lower() == name[-1].lower():

print("yes")

break

i += 1

else:

print("no")

**4ex.py:**

n = int(input("Введите сумму n: "))

bills = [64, 32, 16, 8, 4, 2, 1]

used\_bills = []

for bill in bills:

while n >= bill:

used\_bills.append(bill)

n -= bill

print("Купюры для формирования суммы:", used\_bills)