Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский  
политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчёт по лабораторной работе № 2  
тема «Ветвящиеся алгоритмы и циклы»  
по дисциплине «Информатика»

Выполнил: ПМ-23-2б Воронин М.А.

Проверил: ст. пр. каф. ВММБ Ильиных Г.В.

Пермь, 2023

**Задание 1**

* 1. ***Постановка задачи***

Написать функцию, которая определяет, является ли число простым. Написать функцию, которая определяет, является ли число кубом целого числа. Создать программу, которая в бесконечном цикле получает

от пользователя числа и проверяет их с помощью двух функций.

***1.2 Алгоритм выполнения***

**

*Рис.1 блок-схем задания №1*

***1.3Код программы***

def task1():  
 k = 0  
 while True:  
 a = float(input('Введите значение'))  
 if a <= 0:  
 print(a, 'Не является кубом числа')  
 elif a \*\* (1 / 3) == int(a \*\* (1 / 3)):  
 print(a, 'Является кубом числа')  
 else:  
 print(a, 'Не являетcя кубом числа')  
 for i in range(2, int(a) + 1):  
 if a % i == 0:  
 k += 1  
 if k == 1:  
 print(a, 'Является простым число')  
 else:  
 print(a, 'Не является простым число')  
 break

* 1. ***Тестирование работы программы с проверкой***

Для проверки решений в MS Excel была создана таблица:

A1 – исходное значение

B1 – для нахождения простого или составного числа: =ЕСЛИ(A1=1;"простое";ЕСЛИ(СУММ(--(ОСТАТ(A1;СТРОКА(СМЕЩ($A$1;0;0;A1;1)))=0))=2;"простое";"составное"))

С1 – для нахождения квадрата числа =ЕСЛИ(A1^(1/3) = ЦЕЛОЕ(A1^(1/3));ИСТИНА;ЛОЖЬ)

Таблица 1 – Тестирование работы и проверка решений программы №1

|  |  |  |
| --- | --- | --- |
| №п.п | Решение Python | Решение MS Excel |
| 1 | Введите число **203**  203.0 Не являетcя кубом числа  203.0 Не является простым число | |  |  |  | | --- | --- | --- | | x | Простое или составное? | Куб? | | 203 | Составное | Ложь | |
| 2 | Введите число 3  3.0 Не являетcя кубом числа  3.0 Не является простым число | |  |  |  | | --- | --- | --- | | x | Простое или составное? | Куб? | | 3 | Простое | ложь | |
| 3 | Введите число **8**  8.0 Является кубом числа  8.0 Не является простым число | |  |  |  | | --- | --- | --- | | x | Простое или составное? | Куб? | | 8 | Составное | правда | |

**Задание 2**

***2.1 Постановка задачи***

По формуле вычислить значения функции f (x) в диапазоне значений x ∈ [a, b] и представить в виде графика: f(x) =

***2.2 Алгоритм решения***

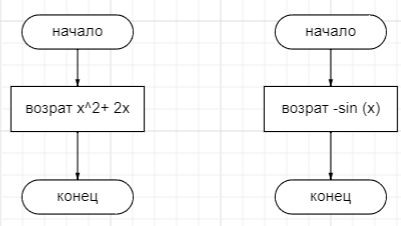
******

Рис.2- Блок-схема *f1(x)* и *f2(x)*

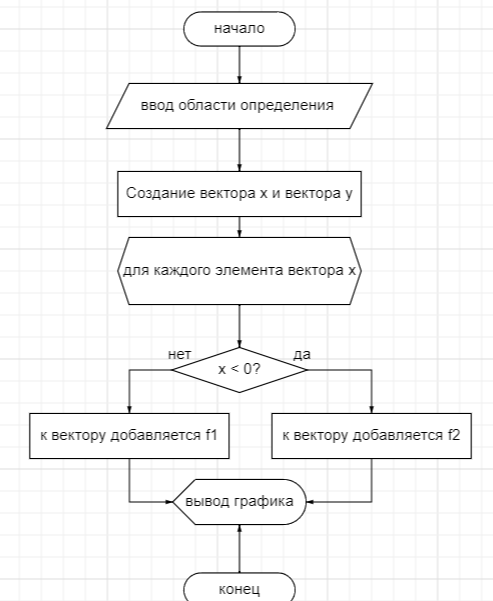


Рис.3 – Блок-схема программы №2

***2.3 Код программы***

def task2():  
 def f1(x):  
 return x \*\* 2 + 2 \* x  
  
 def f2(x):  
 return -m.sin(x)  
  
 obl\_y = (int(input("введите область определения функции \n ")), int(input()))  
 zn\_x = n.linspace(obl\_y[0], obl\_y[1], 500)  
 zn\_y = []  
 for elem in zn\_x:  
 if elem >= 0:  
 zn\_y.append(f1(elem))  
 else:  
 zn\_y.append(f2(elem))  
 plt.plot(zn\_x, zn\_y)  
 plt.grid()  
 plt.show()  
task2()

***2.4 Тестирование работы программы с проверкой***

Для проверки построения графика используется сайт Wolfram Alpha.

Таблица 2 - Тестирование работы и проверка решений программы №2

|  |  |  |
| --- | --- | --- |
| № | Решение Python | График Wolfram Alpha |
| 1 | введите область определения функции **-7 7** |  |
| 2 | введите область определения функции  **-21 21** |  |

**Задание 3  
*3.1 Постановка задачи***

Написать функцию number\_in\_new\_numeral\_system (number, base), которая переводит целое десятичное число из десятичной системы счисления в систему счисления с выбранным основанием. Написать программу, которая переводит заданное пользователем число в заданную пользователем систему счисления.

***3.2 Алгоритм решения***

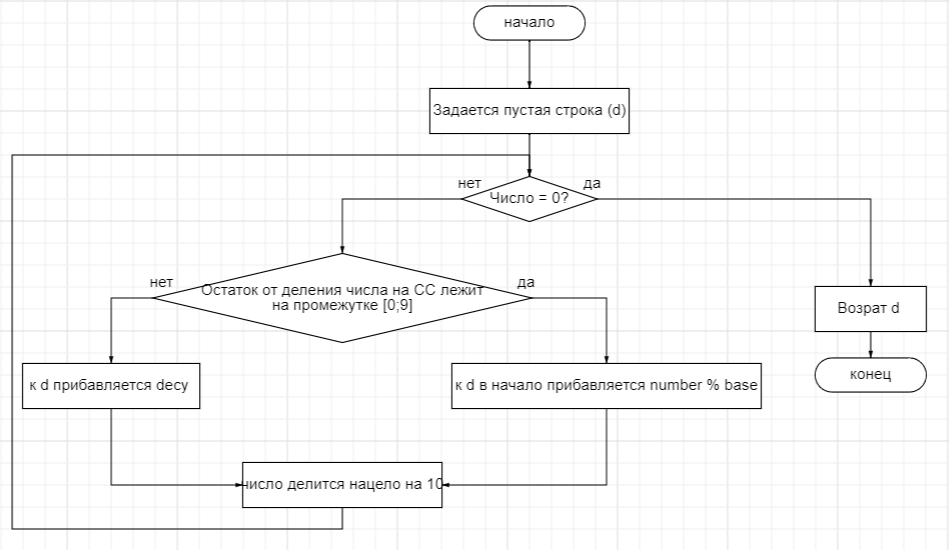


Рис.4 – Блок-схема функции *number\_in\_new\_numeral\_system (number, base)*

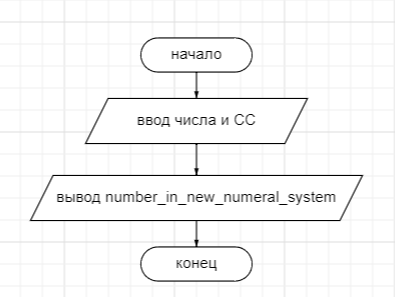


Рис.5 – Блок-схема программы №3

***3.3 Код программы***

def task3():  
 def number\_in\_new\_numeral\_system(number, base):  
 d = ''  
 decy = ["a", "b", "c", "d", "e", "f"]  
 while number > 0:  
 if 0 <= number % base <= 9:  
 d = str(number % base) + d  
 else:  
 d = decy[(number % base) % 10] + d  
 number //= base  
 return d  
 number = int(input("Введите число = "))  
 osn = int(input("Введите основание = "))  
 print(number\_in\_new\_numeral\_system(number, osn))

***3.4 Тестирование работы программы с проверкой***

Для проверки решений используется онлайн-калькулятор систем счисления.

Таблица 3 - Тестирование работы и проверка решений программы №3

|  |  |  |
| --- | --- | --- |
| №п.п | Решение Python | Решение онлайн-калькулятора |
| 1 | Введите число = 15  Введите основание = 16  F |  |
| 2 | Введите число = 16165  Введите основание = 11  11166 |  |
| 3 | Введите число = 715  Введите основание = 3  222111 |  |

**Задание 4**

***4.1 Постановка задачи***

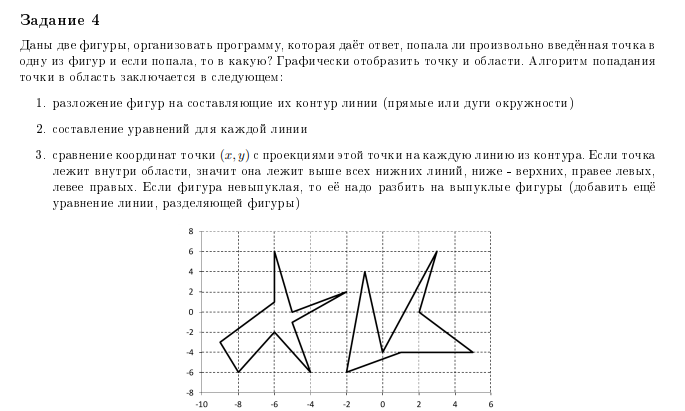


Рис.5 – Задача №4

***4.2 Алгоритм решения***

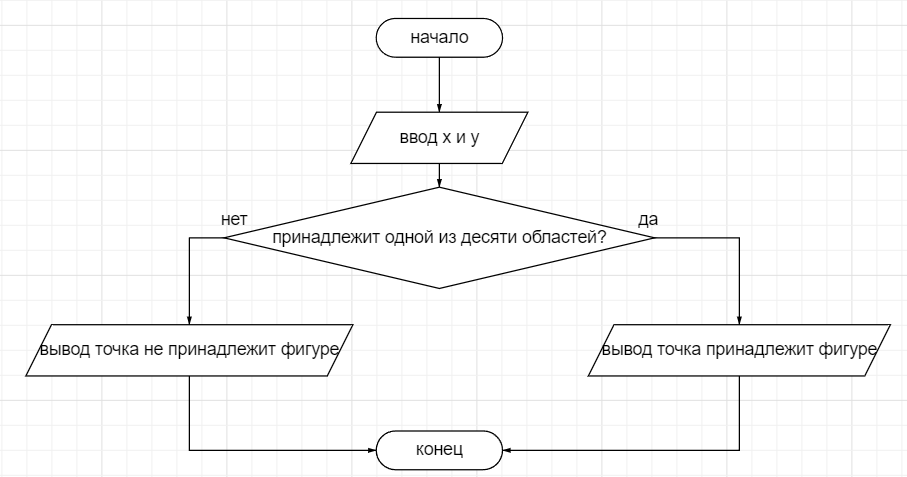


Рис.6 – Блок-схема программы №4

***4.3 Код программы***

def task4():  
 x = float(input('Введите значение x = '))  
 y = float(input('Введите значение y = '))  
 one = ((y <= (4 / 3) \* x + 9) and (y >= -3 \* x - 30) and (y >= 3.5 \* x + 22))  
 two = ((y <= 3.5 \* x + 22) and (y >= 2 \* x + 10) and (x <= -6))  
 there = ((x <= -6) and (y >= -6 \* x - 30) and (y >= 2 \* x + 10))  
 four = ((y <= 2 \* x + 10) and (y >= x + 4) and (x <= -5))  
 five = ((x >= -5) and (y <= (2 / 3) \* x + (10 / 3)) and (y >= x + 4))  
 six = ((y <= x + 4) and (y >= -2 \* x - 14) and (y <= -5 \* x - 26))  
 seven = ((y <= 10 \* x + 14) and (y <= -8 \* x - 4) and (y >= x - 4))  
 eight = ((y <= x - 4) and (y >= (2 / 3) \* x - (14 / 3)) and (y <= - 4))  
 nine = ((y <= (10 / 3) \* x - 4) and (y >= 6 \* x - 12) and (y >= 2 \* x - 4))  
 ten = ((y <= 2 \* x - 4) and (y <= -(4 / 3) \* x + (8 / 3)) and (y >= -4))  
 figure = one or two or there or five or four or six or seven or eight or nine or ten  
 print(f'точка лежит в области - {figure}')

**Задание 5**

***5.1 Постановка задачи***

Написать программу, выводящую на экран те числа, которые больше k и сумма цифр которых больше l. Изначально задается три трехзначных числа.

***5.2 Алгоритм решения***

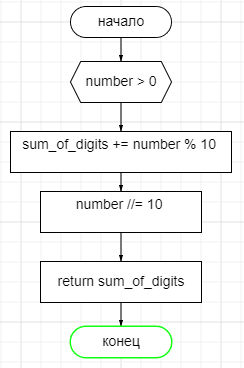


Рис.7 – Блок-схема программы calculate\_digit\_sum

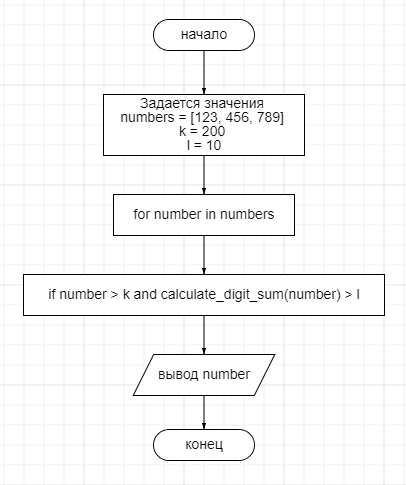


Рис.8 – Блок-схема программы №5

***5.3 Код программы***

def calculate\_digit\_sum(number):  
 sum\_of\_digits = 0  
 while number > 0:  
 sum\_of\_digits += number % 10  
 number //= 10  
 return sum\_of\_digits  
def print\_numbers\_greater\_than\_k\_with\_digit\_sum\_greater\_than\_l(numbers, k, l):  
 for number in numbers:  
 if number > k and calculate\_digit\_sum(number) > l:  
 print(number)  
numbers = [123, 456, 789]  
k = 200  
l = 10  
print\_numbers\_greater\_than\_k\_with\_digit\_sum\_greater\_than\_l(numbers, k, l)

**Задание 6**

***6.1 Постановка задачи***

Известно, что каждую неделю количество травоядных особей и количество хищников увеличивается на 10%. Изначально количество травоядных herbivores, а количество хищников predators. Каждый месяц от старости умирает 1% травоядных и 3% хищников. Учесть, что хищники сокращают каждый месяц популяцию травоядных на 10% от числа хищников. Написать программу, которая определяет размер популяций через заданное число недель, и каждую неделю выводит сообщение о текущей размерности популяций. Добавить условия на снижение численности популяций до нуля.

***6.2 Алгоритм решения***

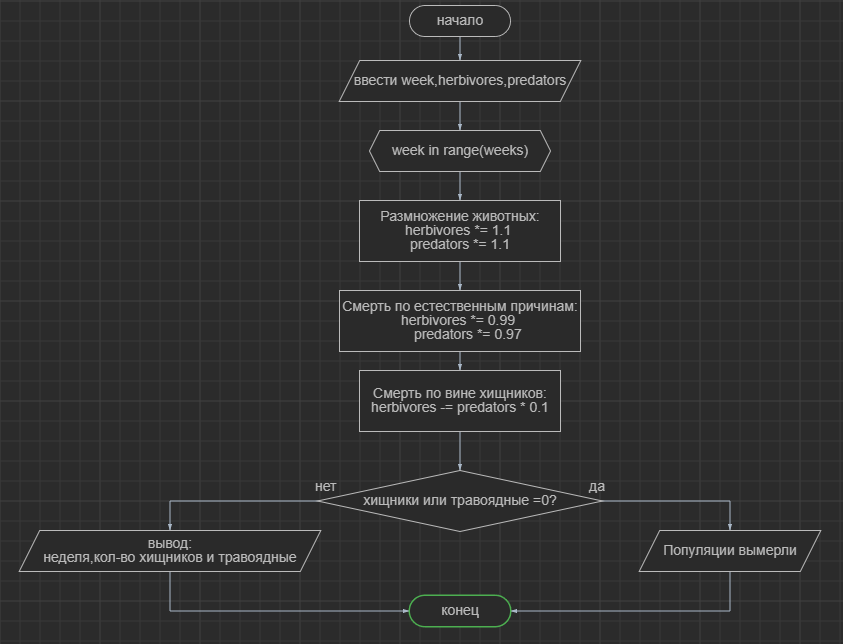


Рис.9 – Блок-схема программы №6

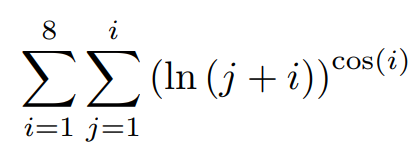
***6.3 Код программы***

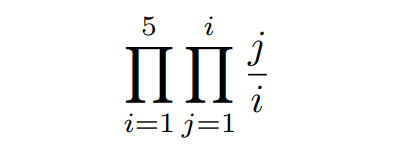
def task6():  
 herbivores = int(input('Введите кол-во травоядных'))  
 predators = int(input('Введите кол-во хищников'))  
 weeks = int(input('Введите число недель'))  
  
 for week in range(weeks):  
 herbivores \*= 1.1  
 predators \*= 1.1  
  
 herbivores \*= 0.99  
 predators \*= 0.97  
  
 herbivores -= predators \* 0.1  
  
 if herbivores <= 0 or predators <= 0:  
 print("Популяции вымерли.")  
 else:  
 print(f'Неделя {week + 1}: Травоядные - {round(herbivores)}, Хищники - {round(predators)}')

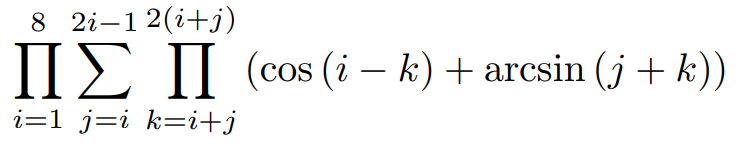
**Задание 8**

***8.1 Постановка задачи***

Посчитать значения выражений:







***8.2 Алгоритм решения***

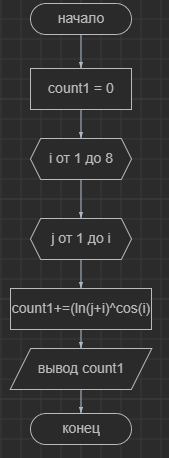
******

Рис.10 – Блок-схема программы №8.1

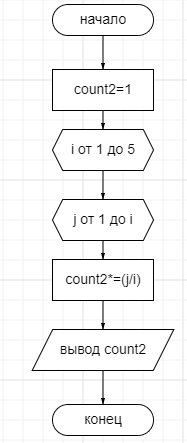
******

Рис.11 – Блок-схема программы №8.2

***8.3 Код программы***

def task8():  
 count1 = 0  
 for i in range(1, 9):  
 for j in range(1, i + 1):  
 count1 = (m.log(2.7, i + j)) \*\* m.cos(i)  
 print(f"Значение первого выражения равно {count1}")  
 count2=1  
 for i in range(1, 6):  
 for j in range(1, i + 1):  
 count2 \*=(j/i)  
 print(f"Значение второго выражения равно {count2}")

***8.4*** ***Тестирование работы программы с проверкой***

Для проверки решения используется сайт Wolfram Alpha и таблица Excel

Таблица 4 - Тестирование работы и проверка решений программы №8

|  |  |  |
| --- | --- | --- |
|  | Решение Python | Решение Wolfram Alpha и Excel |
| 1 | Значение первого выражения равно 1.161094888374988 |  |
| 2 | Значение второго выражения равно 0.0004 |  |