Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский  
политехнический университет

Кафедра «Прикладная математика и механика»

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

Выполнил: студент группы ПМ-23-1б Приходько Е.Д..

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

Пермь, 2023

Содержание

[Задание 1 3](#_Toc156059288)

[1.1. Постановка задачи 3](#_Toc156059289)

[1.2. Алгоритм решения 3](#_Toc156059290)

[1.3. Решение задачи, код программы 3](#_Toc156059291)

[1.4. Тестирование работы программы с проверкой 4](#_Toc156059292)

[Задание 2 4](#_Toc156059293)

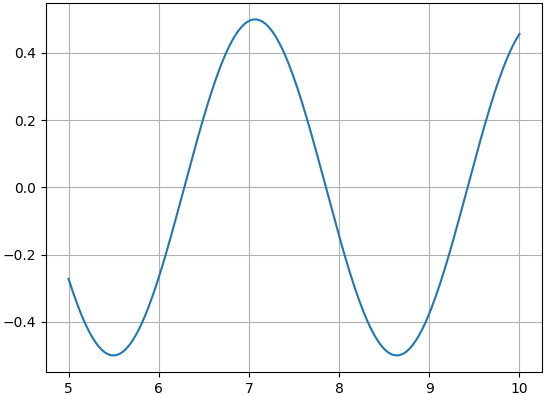
[2.1. Постановка задачи 4](#_Toc156059294)

[2.2. Алгорит решения 5](#_Toc156059295)

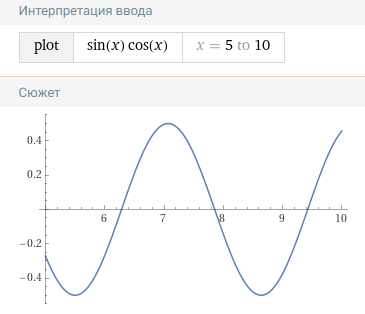
[2.3. Решение задачи, код программы 5](#_Toc156059296)

[2.4. Тестирование работы программы с проверкой 6](#_Toc156059297)

[Python 6](#_Toc156059298)

[****** 6](#_Toc156059299)

[Wolfram 6](#_Toc156059300)

[ 6](#_Toc156059301)

[Задание 3 6](#_Toc156059302)

[3.1. Постановка задачи 6](#_Toc156059303)

[3.2. Алгорит решения 7](#_Toc156059304)

[3.3. Решение задачи, код программы 7](#_Toc156059305)

[3.4. Тестирование работы программы с проверкой 8](#_Toc156059306)

[Задание 4 8](#_Toc156059307)

[4.1. Постановка задачи 8](#_Toc156059308)

[4.2. Алгорит решения 9](#_Toc156059309)

[4.3. Решение задачи, код программы 9](#_Toc156059310)

[4.4. Тестирование работы программы с проверкой 9](#_Toc156059311)

[Задание 5 9](#_Toc156059312)

[5.1. Постановка задачи 9](#_Toc156059313)

[5.1. Алгорит решения 10](#_Toc156059314)

[5.3. Решение задачи, код программы 10](#_Toc156059315)

[5.4. Тестирование работы программы с проверкой 11](#_Toc156059316)

[Задание 6 11](#_Toc156059317)

[6.1. Постановка задачи 11](#_Toc156059318)

[6.1. Алгорит решения 12](#_Toc156059319)

[6.3. Решение задачи, код программы 12](#_Toc156059320)

[6.4. Тестирование работы программы с проверкой 13](#_Toc156059321)

[Задание 8 13](#_Toc156059322)

[8.1. Постановка задачи 13](#_Toc156059323)

[8.2. Алгоритм решения 14](#_Toc156059324)

[8.3. Решение задачи, код программы 14](#_Toc156059325)

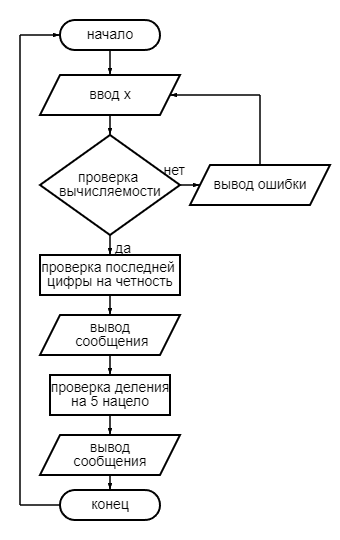
[8.4. Тестирование работы программы с проверкой 15](#_Toc156059326)

Задание 1

1.1. Постановка задачи

*Написать функцию, которая определяет, оканчивается ли число на четную цифру. Написать функцию, которая определяет, делится ли число на пять нацело. Создать программу, которая в бесконечном цикле получает от пользователя числа и проверяет их с помощью двух функций*

1.2. Алгоритм решения



1.3. Решение задачи, код программы

def task1():  
 def f1(x):  
 return (x % 10) % 2 == 0  
 def f2(x):  
 return (x % 5) == 0  
 while True:  
 while True:  
 try:  
 x = int(input("x: "))  
 break  
 except ValueError:  
 print("Введите чиcло")  
 if f1(x):  
 print("Последняя цифра четная")  
 else:  
 print("Последняя цифра нечетная")  
 if f2(x):  
 print("Нацело делится на 5")  
 else:  
 print("Нацело не делится на 5")  
task1()

1.4. Тестирование работы программы с проверкой

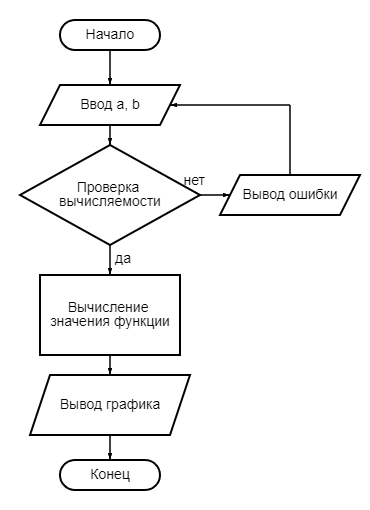
|  |  |
| --- | --- |
| Python |  |
| Ручная проверка | 5%2=2,5 нечетная цифра  5%5=1 нацело делится на 5  2%2=1 четная цифра  2%5=0,4 нацело не делится на 5 |

Задание 2

2.1. Постановка задачи

По формуле вычислить значения функции f(х) в диапазоне х ϵ [a, b] и представить в виде графика.

2.2. Алгорит решения



2.3. Решение задачи, код программы

def task2():  
 import numpy as np  
 import matplotlib.pyplot as pl  
 def f1(x):  
 return np.sin(x)\*np.cos(x)  
 def f2(x):  
 return 2\*x\*\*2+x  
 try:  
 a = (int(input("a: ")))  
 except ValueError:  
 a = (int(input("Введите число: ")))  
 try:  
 b = (int(input("b: ")))  
 except ValueError:  
 b = (int(input("Введите число: ")))  
 x = np.linspace(a, b, 200)  
 y = []  
 for elem in x:  
 if elem >= 0:  
 y.append(f1(elem))  
 else:  
 y.append(f2(elem))  
 pl.plot(x,y)  
 pl.grid()  
 pl.show()  
task2()

2.4. Тестирование работы программы с проверкой

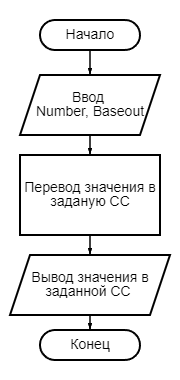
|  |  |
| --- | --- |
| Python |  |
| Wolfram  Alpha |  |

Задание 3

3.1. Постановка задачи

Написать функцию decimnal\_new\_numeral\_system(number, base), которая переводит десятичную дробь из десятичной системы счисления в систему счисления с выбранным основанием. Написать программу, которая переводит пользователем дробь в заданую пользователем систему счисления.

3.2. Алгорит решения

******

3.3. Решение задачи, код программы

def task3():  
 def toBASEint(num, base):  
 alpha = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"  
 n = abs(num)  
 b = alpha[n % base]  
 while n >= base:  
 n = n // base  
 b += alpha[n % base]  
 return ('' if num >= 0 else '-') + b[::-1]  
 def toBaseFrac(frac, base, n=16):  
 alpha = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"  
 b = ''  
 while n:  
 frac \*= base  
 frac = round(frac, n)  
 b += str(alpha[int(frac)])  
 frac -= int(frac)  
 n -= 1  
 return b  
 Number = input("Число: ")  
 Basein = 10  
 Baseout = int(input("Введите СС: "))  
 alpha = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"  
 if '.' in Number:  
 num, frac = map(str, Number.split('.'))  
 num = int(num, Basein)  
 a = toBASEint(num, Baseout)  
 b = 0  
 k = Basein  
 for i in frac:  
 b += alpha.index(i) / k  
 k \*= Basein  
 b = str(toBaseFrac(b, Baseout)).rstrip('0')  
 print("Итог: ",a + '.' + b)  
 else:  
 print("Итог: ",toBASEint(int(Number, Basein), Baseout))  
task3()

3.4. Тестирование работы программы с проверкой

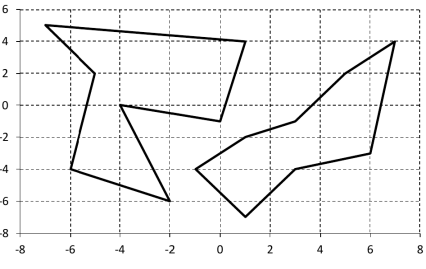
|  |  |
| --- | --- |
| Python |  |
| Калькулятор СС |  |

Задание 4

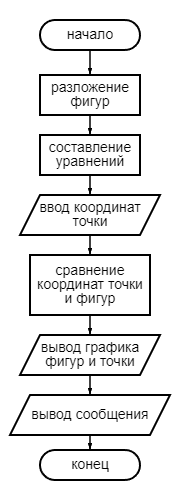
4.1. Постановка задачи

Даны две фигуры, организовать программу, которая дат ответ, попала ли произваольно введенная точка в одну из фигур и если попала, то в какую? Графически отобразить точку и области. Алгоритм попадания точки в область заключается в следующем:

1. Разложение фигур на составляющие их контур линии (прямые или дуги окружности)
2. Составление уравнений для каждой линии
3. Сравнение координат точки (x, y) с проекциями этой точки на каждую линию из контура. Если точка лежит внутри области, значит она лежит выше всех нижних линий, ниже – верхних, правее левых, левее правых. Если фигура невыпуклая, то ее надо разбить на выпуклые фигуры (добавить еще уравнение линии, разделяющей фигуры)



4.2. Алгорит решения



4.3. Решение задачи, код программы

def task4():  
 import matplotlib.pyplot as plt  
 import numpy as np  
  
 def is\_point\_inside\_figure(x, y, point\_x, point\_y):  
 n = len(x)  
 inside = False  
  
 for i in range(n):  
 j = (i + 1) % n  
 if ((y[i] > point\_y) != (y[j] > point\_y)) and \  
 (point\_x < (x[j] - x[i]) \* (point\_y - y[i]) / (y[j] - y[i]) + x[i]):  
 inside = not inside  
  
 if inside:  
 return "Точка находится внутри фигуры"  
 else:  
 return "Точка не находится внутри фигуры"  
  
 x\_coordinates\_shape1 = [-2, -6, -5, -7, 1, 0, -4]

y\_coordinates\_shape1 = [-6, -4, 2, 5, 4, -1, 0]

x\_coordinates\_shape2 = [1, -1, 1, 3, 5, 7, 6, 3]  
 y\_coordinates\_shape2 = [-7, -4, -2, -1, 2, 4, -1, -4]  
  
 test\_point\_x = float(input("Введите координату X контрольной точки: "))  
 test\_point\_y = float(input("Введите координату Y контрольной точки: "))  
  
 # Построение обеих фигур и заданной пользователем контрольной точки  
 plt.plot(x\_coordinates\_shape1 + [x\_coordinates\_shape1[0]], y\_coordinates\_shape1 + [y\_coordinates\_shape1[0]],  
 marker='o', linestyle='-', label='Фигура 1')  
 plt.plot(x\_coordinates\_shape2 + [x\_coordinates\_shape2[0]], y\_coordinates\_shape2 + [y\_coordinates\_shape2[0]],  
 marker='o', linestyle='-', label='Фигура 2')  
 plt.scatter(test\_point\_x, test\_point\_y, color='red', label='Заданная точка')  
  
 # Выделяет выпуклую оболочку каждой фигуры.  
 plt.fill(x\_coordinates\_shape1 + [x\_coordinates\_shape1[0]], y\_coordinates\_shape1 + [y\_coordinates\_shape1[0]],  
 alpha=0.2)  
 plt.fill(x\_coordinates\_shape2 + [x\_coordinates\_shape2[0]], y\_coordinates\_shape2 + [y\_coordinates\_shape2[0]],  
 alpha=0.2)  
  
 plt.xlabel('X-ось')  
 plt.ylabel('Y-ось')  
 plt.title('Фигуры с заданной пользователем точкой')  
 plt.legend()  
 plt.grid(True)  
 plt.show()  
  
 # Проверяет, находится ли заданная пользователем контрольная точка внутри какой-либо из фигур.  
 result\_shape1 = is\_point\_inside\_figure(x\_coordinates\_shape1, y\_coordinates\_shape1, test\_point\_x, test\_point\_y)  
 result\_shape2 = is\_point\_inside\_figure(x\_coordinates\_shape2, y\_coordinates\_shape2, test\_point\_x, test\_point\_y)  
  
 print(result\_shape1)  
 print(result\_shape2)  
  
task4()

4.4. Тестирование работы программы с проверкой

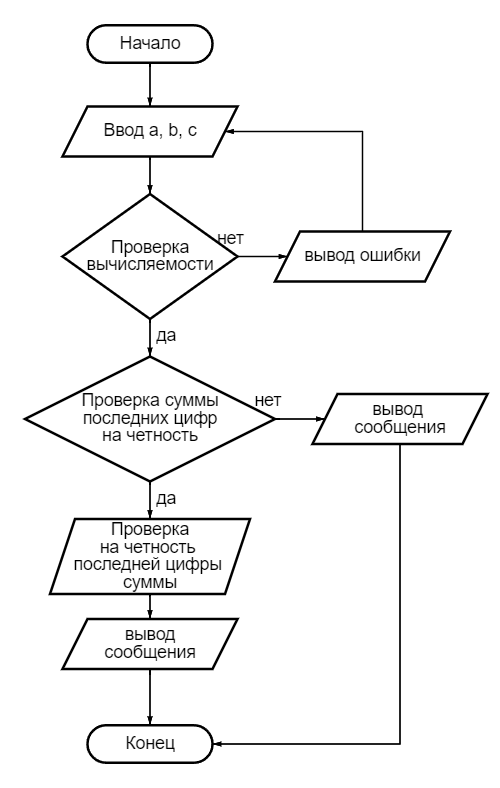
|  |  |
| --- | --- |
| Python |  |
| Ручная проверка |  |

Задание 5

5.1. Постановка задачи

Написать программу, которая проверяет – является ли сумма последних цифр трех заданных чисел четным числом, если да, то проверить на четность последнюю цифру суммы.

5.1. Алгорит решения

******

5.3. Решение задачи, код программы

def task5():  
 try:  
 a = int(input("Введите первое число: "))  
 except ValueError:  
 a = int(input("Введите первое ЧИСЛО: "))  
 try:  
 b = int(input("Введите второе число: "))  
 except ValueError:  
 b = int(input("Введите второе ЧИСЛО: "))  
 try:  
 c = int(input("Введите третье число: "))  
 except ValueError:  
 c = int(input("Введите третье ЧИСЛО: "))  
 a1 = abs(a) % 10  
 b1 = abs(b) % 10  
 c1 = abs(c) % 10  
 sum = a1+b1+c1  
 if sum % 2 == 0:  
 sum1 = abs(sum)%10  
 if sum1 % 2 == 0:  
 print("Число четное")  
 else:  
 print("Число нечетное")  
 else:  
 print("Сумма нечетная")  
task5()

5.4. Тестирование работы программы с проверкой

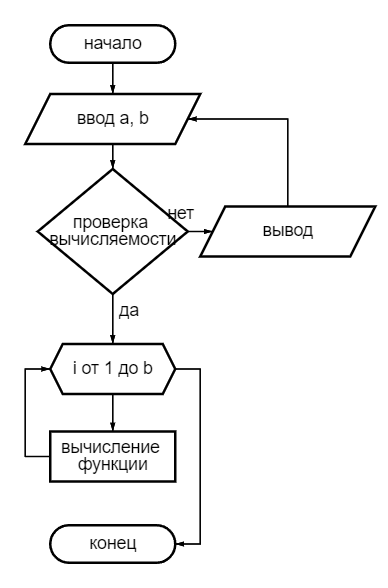
|  |  |
| --- | --- |
| Python |  |
| Ручная проверка | 5+2+7=14  4%2=2 цифра четная |

Задание 6

6.1. Постановка задачи

Каждая бактерия делится на две за одну минуту. В начальный момент start\_amount бактерий. Написать функцию bakterium(amount, minutes), которая возвращает количество бактерий через minutes минут в зависимости от заданного числа бактерий amount. Написать программу, которая с помощью bacterium определяет количество бактерий через заданное время в минутах.

6.1. Алгорит решения



6.3. Решение задачи, код программы

def task6():  
 try:  
 a = int(input("Введите количество бактерий:"))  
 except ValueError:  
 a = int(input("Введите число:"))  
 try:  
 b = int(input("Введите время деления: "))  
 except ValueError:  
 b = int(input("Введите число: "))  
 a1 = a  
 for i in range(0,b):  
 a2 = a1\*2  
 print("Количество бактерий: ",a2)  
 a1 = a2  
task6()

6.4. Тестирование работы программы с проверкой

|  |  |
| --- | --- |
| Python |  |
| Ручная  проверка | 42\*2=84  84\*2=168  168\*2=336  336\*2=672  672\*2=1344 |

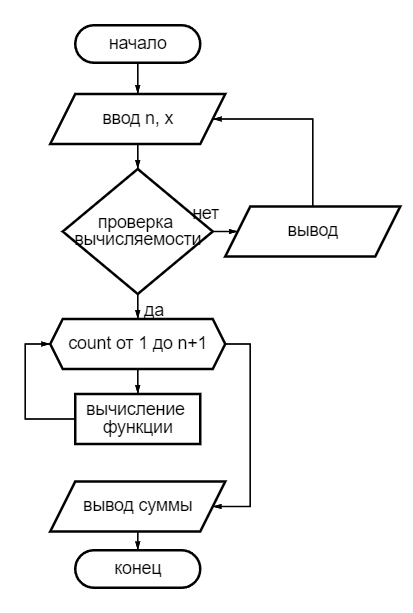
Задание 8

8.1. Постановка задачи

Дан ряд:

Посчитать сумму первых n членов ряда

8.2. Алгоритм решения



8.3. Решение задачи, код программы

def task8():  
 from math import factorial  
 try:  
 n = int(input("Введите длину последовательности: "))  
 except ValueError:  
 n = int(input("Введите число: "))  
 try:  
 x = int(input("Введите x: "))  
 except ValueError:  
 x = int(input("Введите число: "))  
 subcount = 0  
 s = 0  
 for count in range(1, n + 1):  
 s += (factorial(count - 1) / factorial(subcount + count)) \* (x \*\* subcount + count)  
 subcount += 1  
 print(f"Cумма последовательности при x = {x} и длине {n} равна {format(s,'.2f')}")  
task8()

8.4. Тестирование работы программы с проверкой

|  |  |
| --- | --- |
| Python |  |
| Mathway |  |