

ФГБОУ ВО «Московский Политехнический университет»

Лабораторная работа №3
Линейные программы
Задание 1,2,3
Вариант №1

По дисциплине:
Основы Программирования

Выполнил Шукуров Ф.Ф группа 181-362

Проверил _____ Никишина И.Н.

Москва 2018

Задание №1

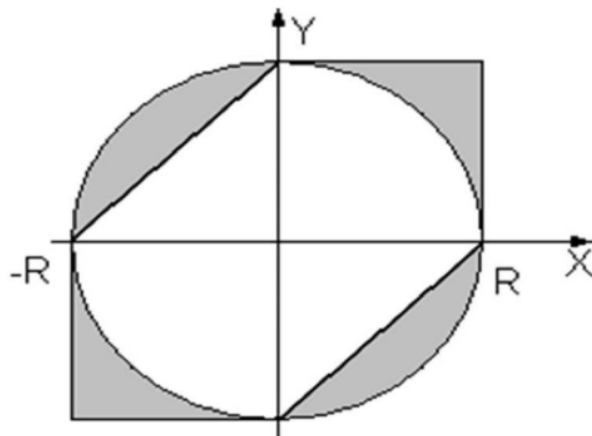
Постановка задачи: Вычислить и вывести на экран или в файл в виде таблицы значения функции, заданной графически (см. лабораторная работа №2, задание 1), на интервале от $X_{нач}$ до $X_{кон}$ шагом dx . Интервал и шаг задавать таким образом, чтобы покрывать все ветки программы. Таблицу снабдить заголовком и шапкой

Описание программы:

Программа была написана на алгоритмическом языке python v3.6, реализованна в среде os Linux, и состоит из блоков ввода, проверки информации и вывода результата. Использован импорт random для вывода «Псевдо случайного числа»

Описание Алгоритма:

1. Объявляем две переменные «xBeg», «xEnd», «dx», присваиваем их к значению float
2. «xt» будет ссылаться на данные «xBeg».
3. Выводим шапку программы
4. Запускаем цикл While с условием « $x \leq xEnd$ »
5. Используя методы исключения (реализованным в python блок if \rightarrow elif \rightarrow else) находим «xt»
6. Выводим информацию на экран
7. Добавляем к «xt» шаг «dx»
8. Как только «xt» становится $\leq xEnd$, цикл While прекращается



ЗАМЕЧАНИЕ: Цикл WHILE МОЖЕТ ПРИНИМАТЬ ЗНАЧЕНИЕ TRUE ИЛИ FALSE

Листинг Программы:

```
# -*- coding: utf-8 -*-
from math import *
import random, math
xBeg = float(input("xBeg = "))
xEnd = float(input("xEnd = "))
dx = float(input("dx = "))
print("xBeg={0: 7.2f} xEnd={1: 7.2f}".format(xBeg,xEnd))
```

```

print("  dx={0:7.2f}".format(dx))
xt=xBeg
print("+-----+-----+")
print("I      X      I      Y      I")
print("+-----+-----+")
while xt<=xEnd:
    if xt <=-2.5: y = -6/7*xt-36/7
    elif -2.5<xt and xt<2: y = xt**3 + 1.5*xt**2-2.5*xt-3
    elif 2<=xt: y = -2*xt + 10
    print("I{0: 7.2f} I{1: 7.2f} I".format(xt,y))
    xt+=dx

```

Результат работы программы:

```

xBeg = 12
xEnd = 24
dx = 1
xBeg= 12.00 xEnd= 24.00
dx= 1.00
+-----+-----+
I      X      I      Y      I
+-----+-----+
I 12.00 I -14.00 I
I 13.00 I -16.00 I
I 14.00 I -18.00 I
I 15.00 I -20.00 I
I 16.00 I -22.00 I
I 17.00 I -24.00 I
I 18.00 I -26.00 I
I 19.00 I -28.00 I
I 20.00 I -30.00 I
I 21.00 I -32.00 I
I 22.00 I -34.00 I
I 23.00 I -36.00 I
I 24.00 I -38.00 I
+-----+-----+

```

Задание№2

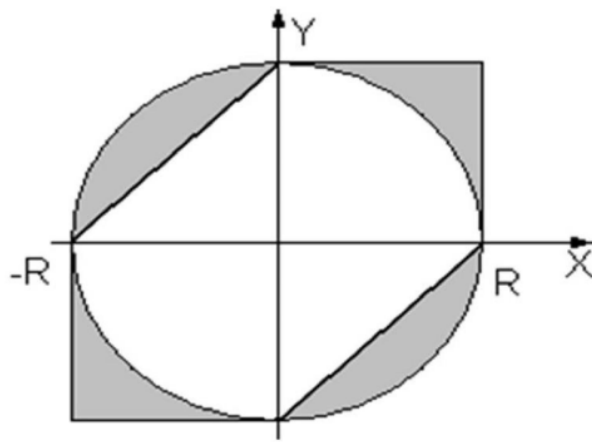
Постановка задачи: Для десяти выстрелов, координаты которых задаются генератором случайных чисел, вывести текстовые сообщения о попадании в мишень(см.лабораторная работа №2,задание 2)

Описание программы:

Программа была написана на алгоритмическом языке python v3.6, реализованна в среде os Linux, и состоит из блоков ввода, проверки информации и вывода результата. Использован импорт random для вывода «Псевдо случайного числа», а так же функции `def name (,)`, аргумент `global аргумент` позволяет ссылаться на аргументы из глобального пространства в локальное.

Описание Алгоритма:

1. Создаем переменные «t», «f» → присваиваем их к значению `int()`
2. Используя `input()` в цикле `for _ in range(_____)`, мы указываем количество выстрелов.
3. Задавая «RADUIS», «x», «y» – псевдо-случайное число используя ранее импортированную функцию `random.randrange()`
4. Используя блок исключений `if → elif → else`, мы исключаем попадание точек в определенную четверть оси координат, при значении `if True` выполняется вызов функции «two four()» или «one three()», в которых продолжаются дальнейшие вычисления
5. В функциях проверяется принадлежность точки к заштрихованной части графика. В случае истины, переменной «t» прибавляется 1. В случае не истины к переменной «f» прибавляется 1.
6. В случае нестандартных значений, которые не принадлежат графику, используется функция `error()`



ЗАМЕЧАНИЕ: ЦИКЛ WHILE МОЖЕТ ПРИНИМАТЬ ЗНАЧЕНИЕ TRUE ИЛИ FALSE

Листинг программы:

```
def error():
    global f
    f+=1
    print('Мимо')
def true():
    global t
    t+=1
    print('Попал')
def two_four():
    if x>0 and y<0 and y<x-RADUIS:
        true()
    elif (x<0 and y>0) and y>x+RADUIS:
        true()
    else:
        error()
def one_three():
    if x<0 and y<0 and y<-x-RADUIS:
        true()
```

```

elif x>0 and y>0 and y>-x-RADUIS:
    true()
else:
    error()
t = int()
f = int()
for i in range(int(input("Введите количество выстрелов: \n"))):
    RADUIS = random.randrange(100)
    x = random.randrange(100)
    y = random.randrange(100)
    po_r = sqrt(x**2+y**2) # point_radius
    if math.fabs(x)<=RADUIS and math.fabs(y)<=RADUIS:
        if po_r<=RADUIS:
            if (x<0 and y>0) or (y<0 and x>0): #первая и четвертая четверть
                two_four()
            else:
                error()
        elif po_r>=RADUIS:
            if (y>0 and x>0) or (x<0 and y<0):
                one_three()
            else:
                error()
        else:
            error()
    else:
        error()
print("Выстрелов попавших в цель: " + str(t))
print("Выстрелов не попавших в цель: " + str(f))

```

Работа программы:

```

Введите количество выстрелов:
10
Мимо
Мимо
Мимо
Мимо
Попал
Мимо
Мимо
Мимо
Мимо
Мимо
Выстрелов попавших в цель: 1
Выстрелов не попавших в цель: 9

```

Задание №3

Постановка задачи: Вычислять и вывести на экран в виде таблицы значения функции $(1+x)^{-\frac{1}{3}}$ заданной рядом Тейлора \rightarrow

$$1 - \frac{1}{1.2}(2 \cdot 3 \cdot x - 3 \cdot 4 \cdot x^2 + 4 \cdot 5 \cdot x^3 - 5 \cdot 6 \cdot x^4 + \dots) \quad |x| \leq 1$$

на интервале от $X_{нач}$ до $X_{кон}$ с шагом dx с точностью ϵ

Описание программы:

Программа была написана на алгоритмическом языке python v3.6, реализованна в среде os Linux, и состоит из блоков ввода, проверки информации и вывода результата. Использован цикл «While», а так же блоки проверки истинности.

Описание Алгоритма:

1. Создаем бесконечный цикл в бесконечном цикле «While True → While True» для будущей проверки переменной xBeg, которую будет указывать пользователь.
2. Если число введенной пользователем по модулю $|xBeg| \leq 1$, выход из цикла
3. Начинается проверка xEnd
4. Если $|xEnd| \leq 1$, выход из цикла
5. Вывод шапки
6. Создается переменная xt которая ссылается на xBeg ($xt \rightarrow xbeg$)
7. Запускаем цикл с уловием: $xt \leq xEnd$, а так же в теле цикла создаем переменную result1, которая равна 0, а так же n которая равна 2.
8. входим в цикл for, который будет выполнять действие количество раз, указанное пользователем.
9. Проверяем счет цикла на чётность, методом получения остатка переменной i при делении на 2. При получении $0 \rightarrow \text{False}$, $1 \rightarrow \text{True}$. Следовательно, мы прибавляем либо отнимаем от result1 ответ формулы $n \cdot (n + 1) \cdot xt^{(n-1)}$
10. вывод $1 - \frac{result1}{2}$ на экран
11. добавление к xt шаг dx

```
while True:
    while True:
        xBeg = float(input("xBeg[-1:1]= "))
        if abs(xBeg)<=1:
            break
        xEnd = float(input("xEnd[-1:1] = "))
        if abs(xEnd)<=1:
            break
    dx = float(input("dx = "))
    print("xBeg={0: 7.2f} xEnd={1: 7.2f}".format(xBeg,xEnd))
    print("dx={0:7.2f}".format(dx))
    xt=xBeg
    print("+-----+-----+")
    print("I      X      I      Y      I")
    print("+-----+-----+")
    while xt<=xEnd:
        result_1 = 0
        n=2
        for i in range(int(input("Укажите точность расчетов:\t"))):
            if i%2: result_1-= (n*(n+1)*xt**(n-1))
            else:   result_1+= (n*(n+1)*xt**(n-1))
            n+=1
```

```

    print("I{0: 7.2f}  I {1: 7.2f}  I".format(xt, (1-result_1/2)))
    xt+=dx
    result_1 = 0
print("+-----+-----+")

```

Результат работы программы:

```

xBeg[-1:1]= 0.01
xEnd[-1:1] = 1
dx = 0.1
xBeg= 0.01 xEnd= 1.00
dx= 0.10
+-----+-----+
I      X      I      Y      I
+-----+-----+
I  0.01  I    0.97  I
I  0.11  I    0.73  I
I  0.21  I    0.56  I
I  0.31  I    0.44  I
I  0.41  I    0.36  I
I  0.51  I    0.29  I
I  0.61  I    0.24  I
I  0.71  I    0.21  I
I  0.81  I    0.58  I
I  0.91  I   14.56  I
+-----+-----+

```