

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт компьютерных наук и технологий

Высшая школа искусственного интеллекта

Дисциплина:
ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

ПРАКТИЧЕСКОЕ ЗАДАНИЕ 1

Обучающийся гр. 3530201/10001

Нгуен Куок Дат

Руководитель

Моторин Дмитрий Евгеньевич

Санкт-Петербург 2022

Содержание

1	Реализация	2
1.1	Задание 1	2
1.2	Задание 2	3
1.3	Задание 3	3
1.4	Задание 4	4
1.5	Задание 5	4
1.6	Задание 6	5
1.6.1	Теоретическое сведение	5
1.6.2	Реализация	6
2	Результат работы	9

1 Реализация

1.1 Задание 1

Реализовать функцию

$$\prod_{l=1}^n \sum_{k=1}^l \prod_{p=1}^k \sum_{i=1}^p i$$

со значением $n = 3$.

При запуске функции используя команду “:set +s”, записывать в отчет время выполнения и количество затрачиваемой памяти.

Надо реализовать 4 рекурсивных функций: сложение до p **addp: Int->Int**, умножение до k **mulk: Int->Int**, сложение до l **addl: Int->Int**, умножение до n **muln: Int->Int**. Результат задания - вывод функции **muln**

```
addp :: Int->Int
addp 1 = 1
addp p = p + addp (p - 1)

mulk :: Int->Int
mulk 1 = 1
mulk k = addp k * mulk (k - 1)

addl :: Int->Int
addl 1 = 1
addl l = mulk l + addl (l - 1)

muln :: Int->Int
```

```

muln 1 = 1
muln n = add1 n * muln (n - 1)

```

1.2 Задание 2

Реализовать функцию, разворачивающую список $S1 = [kiveydewhjusgofimbbbyhwbopvuplwfexresmhntnic]$ ($[1,2,3] \rightarrow [3,2,1]$).

Реализуют функцию **reverse1: String->String**, в которой рекурсивно перемещают голову строки в конец.

```

reverse1 :: String -> String
reverse1 [] = []
reverse1 a = (++) (reverse1 (tail a)) [head a]

```

1.3 Задание 3

Реализовать функцию конкатенации списка списков $S2 = [[kiv], [eydew], [hjusg], [mbbyhwbopvupl], [ofi], [wfexresmhntnic]]$ ($[[1], [2], [3], [4]] \rightarrow [1,2,3,4]$)

Реализуют функцию **concat1: [String]->String**, в которой рекурсивно добавляют каждую данную строку в одну другую.

```

concat1 :: [String] -> String
concat1 [] = []
concat1 a = (++) (head a) (concat1 (tail a))

```

1.4 Задание 4

Реализовать функцию, которая вставляет строку “-,-” между списками S_1 и S_2 .

Реализуют функцию `separate: [String] -> String`, в которой рекурсивно конкатенируют в порядке элементарную строку, “-,-” в начало результирующей строки.

```
separate:: [String] -> String
separate [] = []
separate a = concat1 [head a, "-,-", separate (tail a)]
```

1.5 Задание 5

Реализовать вычисление элемента треугольника Паскаля (расчет i -ого числа в j -ой строке).

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```

Рис. 1: Треугольник Паскаля

Формула вычисления число Паскаля на строку **j** и столбец **i**:

$$p(0,0) = p(0,i) = p(i,i) = 1$$

$$p(i,j) = p(i-1,j-1) + p(i,j-1)$$

Реализуют функцию **pascal: (Int,Int)->Int**, в которой рекурсивно вычисляют число Паскаля с данным положением.

```
pascal :: Int->Int->Int
pascal 0 0 = 1
pascal 0 a = 1
pascal i j =
    if i == j then 1
    else pascal (i-1) (j-1) + pascal i (j-1)
```

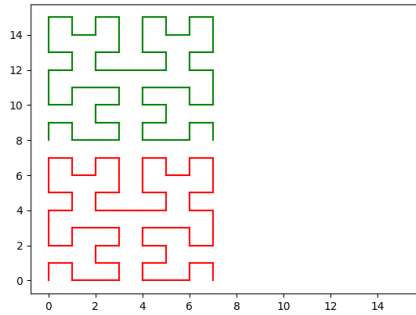
1.6 Задание 6

1.6.1 Теоретическое сведение

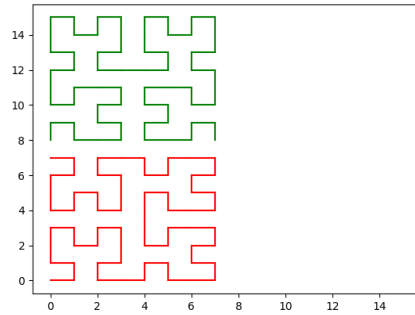
Кривая Гильберта (известная также как заполняющая пространство кривая Гильберта) — это непрерывная фрактальная заполняющая пространство кривая, впервые описанная немецким математиком Давидом Гильбертом в 1891 году, как вариант заполняющих пространство кривых Пеано, открытых итальянским математиком Джузеппе Пеано в 1890 году.

1.6.2 Реализация

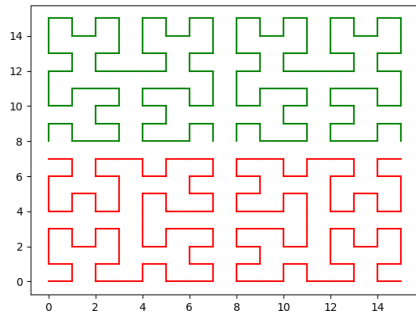
Кривую Гильберта могут рисовать по преобразованию сдвига, поворота и горизонтальной рефлексии.



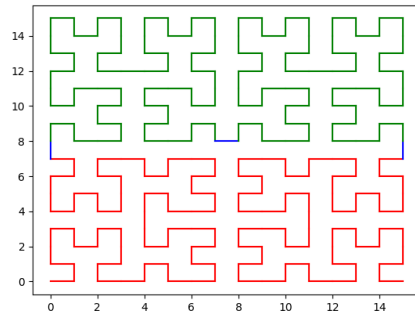
(a) Нарисовать 2 кривые степени $n-1$ на 2-ых и 3-их квадрантах



(b) Повернуть направо нижнюю кривую



(c) Рефлексировать левую половину направо



(d) Соединить начала и концы фигур

Рис. 2: Рекурсивное строение кривой Гильберта ($n=4$)

```

-- rotate right
rotr :: Int -> [[Int]] -> [[Int]]
rotr n [] = []
rotr n a = [last (head a), n - 1 - head(head a)] : rotr n (tail a)

-- 2^n
power2 :: Int -> Int
power2 0 = 1
power2 n = 2 * power2 (n-1)

-- reflex horizontal
reflexy :: Int -> [[Int]] -> [[Int]]
reflexy n [] = []
reflexy n a = [n - 1 - head (last a), last (last a)] : reflexy n
  ↪ (init a)

-- reverse sequence of coordinates (for easier visualization)
rev :: [[Int]] -> [[Int]]
rev [a] = [a]
rev a = (last a) : rev (init a)

-- shift by vertical axis
shiftup :: Int -> [[Int]] -> [[Int]]
shiftup a [] = []

```



```
shiftup a b = [head (head b), a+ last(head b)]: shiftup a (tail b)

--display curve by sequence of coordinates
hilbert:: Int -> [[Int]]
hilbert 0  = []
hilbert 1  = [[0,0],[0,1],[1,1],[1,0]]
```

2 Результат работы

Функция `main`

```
main :: IO ()
main = do
    print "Exercise 1: Result is"
    print(muln 3)
    print "Exercise 2: Reversed list: "
    print (reverse1 "kiveydewhjugofiimbbyhwbopvuplwfexresmhtnic")
    print "Exercise 3: Concatenation: "
    print (concat1 ["kiv", "eydew", "hjug", "mbbyhwbopvupl",
        ↪ "ofii", "wfexresmhtnic"])
    print "Exercise 4: "
    print (separate ["kiv", "eydew", "hjug", "mbbyhwbopvupl",
        ↪ "ofii", "wfexresmhtnic"])
    print "Exercise 5: Pascal number "
    print (pascal 3 4)
    print "Exercise 6: Hilbert curve 4 iterations:"
    print (hilbert 4)
```

```

F:\git\FuncProg\Assignment1>ghci
GHCi, version 9.2.5: https://www.haskell.org/ghc/  :? for help
ghci> :l Assign1
[1 of 1] Compiling Main                ( Assign1.hs, interpreted )
Ok, one module loaded.
ghci> main
"Exercise 1: Result is"
88
"Exercise 2: Reversed list: "
"cinthmserxewflpvpobwhybbmiifogsujhwedyevik"
"Exercise 3: Concatenation: "
"kiveydewhjsgmbbyhwbopvuplofiwfwexresmhtnic"
"Exercise 4: "
"kiv-, -eydew-, -hjusg-, -mbbyhwbopvupl-, -ofii-, -wfwexresmhtnic-, -"
"Exercise 5: Pascal number "
4
"Exercise 6: Hilbert curve 4 iterations:"
[[0,0],[1,0],[1,1],[0,1],[0,2],[0,3],[1,3],[1,2],[2,2],[2,3],[3,3],[3,2],[3,1],[2,1],[2,0],[3,0],[4,0],[4,1],[5,1],
[5,0],[6,0],[7,0],[7,1],[6,1],[6,2],[7,2],[7,3],[6,3],[5,3],[5,2],[4,2],[4,3],[4,4],[4,5],[5,5],[5,4],[6,4],[7,4],
[7,5],[6,5],[6,6],[7,6],[7,7],[6,7],[5,7],[5,6],[4,6],[4,7],[3,7],[2,7],[2,6],[3,6],[3,5],[3,4],[2,4],[2,5],[1,5],
[1,4],[0,4],[0,5],[0,6],[1,6],[1,7],[0,7],[0,8],[0,9],[1,9],[1,8],[2,8],[3,8],[3,9],[2,9],[2,10],[3,10],[3,11],[2,11],
[1,11],[1,10],[0,10],[0,11],[0,12],[1,12],[1,13],[0,13],[0,14],[0,15],[1,15],[1,14],[2,14],[2,15],[3,15],[3,14],
[4,13],[2,13],[2,12],[3,12],[4,12],[5,12],[5,13],[4,13],[4,14],[4,15],[5,15],[5,14],[6,14],[6,15],[7,15],[7,14],
[7,13],[6,13],[6,12],[7,12],[7,11],[7,10],[6,10],[6,11],[5,11],[4,11],[4,10],[5,10],[5,9],[4,9],[4,8],[5,8],[6,8],
[6,9],[7,9],[7,8],[8,8],[8,9],[9,9],[9,8],[10,8],[11,8],[11,9],[10,9],[10,10],[11,10],[11,11],[10,11],[9,11],[9,10],
[8,10],[8,11],[8,12],[9,12],[9,13],[8,13],[8,14],[8,15],[9,15],[9,14],[10,14],[10,15],[11,15],[11,14],[11,13],[10,13],
[10,12],[11,12],[12,12],[13,12],[13,13],[12,13],[12,14],[12,15],[13,15],[13,14],[14,14],[14,15],[15,15],[15,14],
[15,13],[14,13],[14,12],[15,12],[15,11],[15,10],[14,10],[14,11],[13,11],[12,11],[12,10],[13,10],[13,9],[12,9],
[12,8],[13,8],[14,8],[14,9],[15,9],[15,8],[15,7],[14,7],[14,6],[15,6],[15,5],[15,4],[14,4],[14,5],[13,5],[13,4],[12,4],
[12,5],[12,6],[13,6],[13,7],[12,7],[11,7],[11,6],[10,6],[10,7],[9,7],[8,7],[8,6],[9,6],[9,5],[8,5],[8,4],[9,4],
[10,4],[10,5],[11,5],[11,4],[11,3],[11,2],[10,2],[10,3],[9,3],[8,3],[8,2],[9,2],[9,1],[8,1],[8,0],[9,0],[10,0],
[10,1],[11,1],[11,0],[12,0],[13,0],[13,1],[12,1],[12,2],[12,3],[13,3],[13,2],[14,2],[14,3],[15,3],[15,2],[15,1],[14,1],
[14,0],[15,0]]
ghci>

```

Рис. 3: Программа выполнена на консоле

Отображение кривых на **Python** с библиотекой **matplotlib**

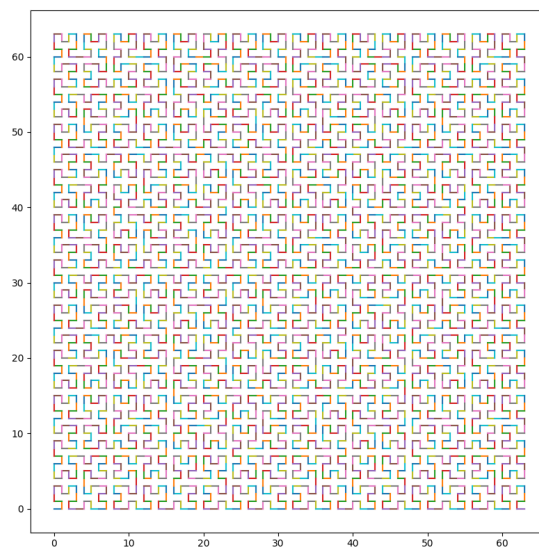
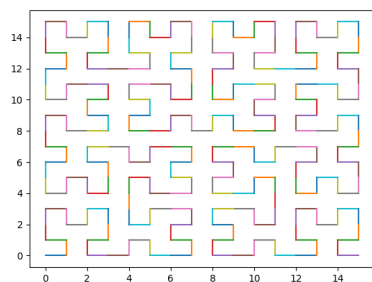


Рис. 4: Кривые Гилберта с 4 и 6 итераций

Вывод