

# Алгоритмизация и программирование лабораторные работы

Иванова Г.С.  
Аргентов П.В.  
ИУ-6

Сентябрь 2021

## Содержание

<b>Общие правила</b>	<b>6</b>
<b>1 Вариант 1</b>	<b>8</b>
1.1 ЛР 1. Базовое структурное программирование . . . . .	8
1.2 ЛР 2. Коллекции и строки . . . . .	9
1.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	10
1.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	11
<b>2 Вариант 2</b>	<b>12</b>
2.1 ЛР 1. Базовое структурное программирование . . . . .	12
2.2 ЛР 2. Коллекции и строки . . . . .	13
2.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	14
2.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	15
<b>3 Вариант 3</b>	<b>16</b>
3.1 ЛР 1. Базовое структурное программирование . . . . .	16
3.2 ЛР 2. Коллекции и строки . . . . .	17
3.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	18
3.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	19
<b>4 Вариант 4</b>	<b>20</b>
4.1 ЛР 1. Базовое структурное программирование . . . . .	20
4.2 ЛР 2. Коллекции и строки . . . . .	21
4.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	22
4.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	23
<b>5 Вариант 5</b>	<b>24</b>
5.1 ЛР 1. Базовое структурное программирование . . . . .	24
5.2 ЛР 2. Коллекции и строки . . . . .	25
5.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	26

5.4	ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	27
<b>6</b>	<b>Вариант 6</b>	<b>28</b>
6.1	ЛР 1. Базовое структурное программирование . . . . .	28
6.2	ЛР 2. Коллекции и строки . . . . .	29
6.3	ЛР 3. Функциональная и модульная декомпозиция . . . . .	30
6.4	ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	31
<b>7</b>	<b>Вариант 7</b>	<b>32</b>
7.1	ЛР 1. Базовое структурное программирование . . . . .	32
7.2	ЛР 2. Коллекции и строки . . . . .	33
7.3	ЛР 3. Функциональная и модульная декомпозиция . . . . .	34
7.4	ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	35
<b>8</b>	<b>Вариант 8</b>	<b>36</b>
8.1	ЛР 1. Базовое структурное программирование . . . . .	36
8.2	ЛР 2. Коллекции и строки . . . . .	37
8.3	ЛР 3. Функциональная и модульная декомпозиция . . . . .	38
8.4	ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	39
<b>9</b>	<b>Вариант 9</b>	<b>40</b>
9.1	ЛР 1. Базовое структурное программирование . . . . .	40
9.2	ЛР 2. Коллекции и строки . . . . .	41
9.3	ЛР 3. Функциональная и модульная декомпозиция . . . . .	42
9.4	ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	43
<b>10</b>	<b>Вариант 10</b>	<b>44</b>
10.1	ЛР 1. Базовое структурное программирование . . . . .	44
10.2	ЛР 2. Коллекции и строки . . . . .	45
10.3	ЛР 3. Функциональная и модульная декомпозиция . . . . .	46
10.4	ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	47
<b>11</b>	<b>Вариант 11</b>	<b>48</b>
11.1	ЛР 1. Базовое структурное программирование . . . . .	48
11.2	ЛР 2. Коллекции и строки . . . . .	49
11.3	ЛР 3. Функциональная и модульная декомпозиция . . . . .	50
11.4	ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	51
<b>12</b>	<b>Вариант 12</b>	<b>52</b>
12.1	ЛР 1. Базовое структурное программирование . . . . .	52
12.2	ЛР 2. Коллекции и строки . . . . .	53
12.3	ЛР 3. Функциональная и модульная декомпозиция . . . . .	54
12.4	ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	55

<b>13 Вариант 13</b>	<b>56</b>
13.1 ЛР 1. Базовое структурное программирование . . . . .	56
13.2 ЛР 2. Коллекции и строки . . . . .	57
13.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	58
13.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	59
<b>14 Вариант 14</b>	<b>60</b>
14.1 ЛР 1. Базовое структурное программирование . . . . .	60
14.2 ЛР 2. Коллекции и строки . . . . .	61
14.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	62
14.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	63
<b>15 Вариант 15</b>	<b>64</b>
15.1 ЛР 1. Базовое структурное программирование . . . . .	64
15.2 ЛР 2. Коллекции и строки . . . . .	65
15.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	66
15.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	67
<b>16 Вариант 16</b>	<b>68</b>
16.1 ЛР 1. Базовое структурное программирование . . . . .	68
16.2 ЛР 2. Коллекции и строки . . . . .	69
16.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	70
16.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	71
<b>17 Вариант 17</b>	<b>72</b>
17.1 ЛР 1. Базовое структурное программирование . . . . .	72
17.2 ЛР 2. Коллекции и строки . . . . .	73
17.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	74
17.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	75
<b>18 Вариант 18</b>	<b>76</b>
18.1 ЛР 1. Базовое структурное программирование . . . . .	76
18.2 ЛР 2. Коллекции и строки . . . . .	77
18.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	78
18.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	79
<b>19 Вариант 19</b>	<b>80</b>
19.1 ЛР 1. Базовое структурное программирование . . . . .	80
19.2 ЛР 2. Коллекции и строки . . . . .	81
19.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	82
19.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	83
<b>20 Вариант 20</b>	<b>84</b>
20.1 ЛР 1. Базовое структурное программирование . . . . .	84

20.2 ЛР 2. Коллекции и строки . . . . .	85
20.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	86
20.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	87
<b>21 Вариант 21</b>	<b>88</b>
21.1 ЛР 1. Базовое структурное программирование . . . . .	88
21.2 ЛР 2. Коллекции и строки . . . . .	89
21.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	90
21.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	91
<b>22 Вариант 22</b>	<b>92</b>
22.1 ЛР 1. Базовое структурное программирование . . . . .	92
22.2 ЛР 2. Коллекции и строки . . . . .	93
22.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	94
22.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	95
<b>23 Вариант 23</b>	<b>96</b>
23.1 ЛР 1. Базовое структурное программирование . . . . .	96
23.2 ЛР 2. Коллекции и строки . . . . .	97
23.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	98
23.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	99
<b>24 Вариант 24</b>	<b>100</b>
24.1 ЛР 1. Базовое структурное программирование . . . . .	100
24.2 ЛР 2. Коллекции и строки . . . . .	101
24.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	102
24.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	103
<b>25 Вариант 25</b>	<b>104</b>
25.1 ЛР 1. Базовое структурное программирование . . . . .	104
25.2 ЛР 2. Коллекции и строки . . . . .	105
25.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	106
25.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	107
<b>26 Вариант 26</b>	<b>108</b>
26.1 ЛР 1. Базовое структурное программирование . . . . .	108
26.2 ЛР 2. Коллекции и строки . . . . .	109
26.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	110
26.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	111
<b>27 Вариант 27</b>	<b>112</b>
27.1 ЛР 1. Базовое структурное программирование . . . . .	112
27.2 ЛР 2. Коллекции и строки . . . . .	113
27.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	114

27.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	115
<b>28 Вариант 28</b>	<b>116</b>
28.1 ЛР 1. Базовое структурное программирование . . . . .	116
28.2 ЛР 2. Коллекции и строки . . . . .	117
28.3 ЛР 3. Функциональная и модульная декомпозиция . . . . .	118
28.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода . . . . .	119

## Общие правила

### Проведение ЛР

1. Посещение лабораторных работ является обязательным. В помещение не допускаются студенты в верхней одежде, с едой и напитками.
2. В случае пропуска лабораторной работы по уважительной причине студент обязан предъявить оправдательный документ (справку о болезни, повестку в военкомат и т. д.)
3. Разрешается пропуск занятий студентам, которые сдали программы и отчеты по всем предыдущим лабораторным работам, включая пропускаемую.
4. Во время занятий студент выполняет лабораторную работу, составляет отчет и показывает работающую программу и отчет преподавателю. Не разрешается в учебное время ходить по помещению, громко разговаривать и играть в компьютерные игры.
5. Студенты, не получившие зачет по всем лабораторным работам, к сдаче экзамена не допускаются. Студентам, сдающим лабораторные работы после срока по неуважительной причине, зачет задачи по результатам рейтинга аннулируется, а оценка за экзамен снижается.
6. Студенты, не выполняющие настоящие правила, от занятий отстраняются и будут допущены до следующих занятий только при наличии объяснительной записки с визой заместителя декана I курса.

### Требования к отчетам по практике

1. Задание по лабораторной работе считается выполненным, если преподаватель принял программу и отчет.
2. Отчет должен выполняться на любой бумаге формата А4 или А5 в том числе в тетрадях или на тетрадных листах. Если отчет выполняется на отдельных тетрадных листах, то они должны быть аккуратно обрезаны по линии подшивки и скреплены. Неаккуратно выполненные, оборванные или грязные отчеты не принимаются.
3. Все записи должны быть либо напечатаны на принтере, либо разборчиво выполнены от руки синей или черной ручкой (карандаш — не допускается). Схемы также должны быть напечатаны при помощи компьютера или нарисованы карандашом (или ручкой) с использованием чертежных инструментов.
4. Каждый отчет должен включать:
  1. Фамилию, имя и номер группы студента;
  2. номер варианта;
  3. номер и название лабораторной работы;
  4. текст задания;
  5. текст программы;
  6. результаты тестирования.

Результаты тестирования должны быть оформлены в виде таблицы вида:

Исходные данные	Ожидаемый результат	Полученный результат

5. Если студент имеет более 2-х несданных работ, то следующие лабораторные у него приниматься не будут, пока предыдущие лабораторные работы не будут сданы, а отчеты по ним — подписаны.

## Среда исполнения учебных программ

Учебные программы, разрабатываемые на лабораторных работах (далее — *программы*), пишутся и исполняются в среде "Julia for VSCode". Среда использует инструментарий Julia, установленный в ОС студента (на лабораторном ПК или собственном компьютере/ноутбуке студента). В качестве запасного варианта при невозможности установить стандартную среду разрешается использовать Google Colab с настройками для исполнения Julia.

Исходный код упражнений нужно сохранять в файлах с суффиксом `.jl` для возможного последующего разбора и удобной интеграции в отчёты.

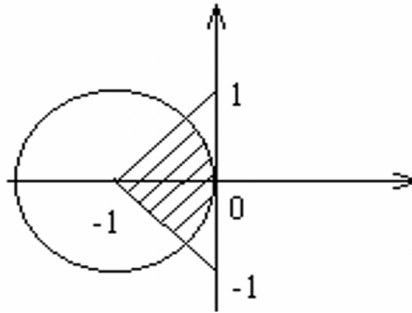
Полезные ссылки:

1. Официальная страница дистрибутивов Julia: <https://julialang.org/downloads/>
2. Официальная страница дистрибутивов VSCode: <https://code.visualstudio.com/download>
3. Официальная страница среды Julia for VSCode: <https://www.julia-vscode.org/>
4. Инструкции по настройке Julia в Google Colab: <https://github.com/Dsantra92/Julia-on-Colab>

# 1 Вариант 1

## 1.1 ЛР 1. Базовое структурное программирование

### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

### Задача

В заданном целом числе определить количество четных цифр.



## 1.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

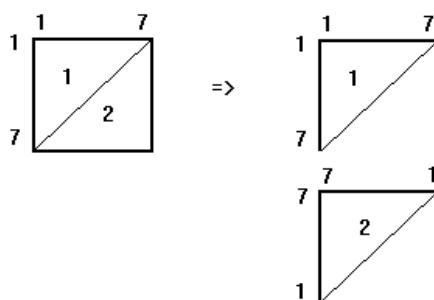
В результате измерений получено  $n \leq 50$  значений некоторой величины  $A$ , которые вводятся с клавиатуры. Написать и отладить программу определения разности между максимальным и минимальным значением измерений. Вывести на экран исходный массив и результат с комментариями.

Исходные данные для отладки:

$n=10$ ,  $A = \{13.249, 8.548, 39.427, -0.813, 5.413, -0.083, 8.293, 9.541, 12.143, 0.519\}$ .

### Задание 2. Матрицы

Дан целочисленный массив размером  $7 \times 7$ , каждый элемент которого – случайно заданные цифры от 0 до 9. Вывести массив на экран в виде, показанном на рисунке:



### Задание 3. Строки

Дана непустая последовательность слов из строчных латинских букв (количество слов больше 3): между соседними словами – запятая, за последним словом – точка. Определить количество слов, которые начинаются и заканчиваются на гласную букву. Вывести на экран количество слов и сами найденные слова.

### 1.3 ЛР 3. Функциональная и модульная декомпозиция

#### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

#### **Задача**

Треугольник задан координатами своих вершин. Кроме того, даны координаты  $m$  точек на плоскости. Определить, сколько точек находится внутри фигуры. Для определения принадлежности точки треугольнику использовать метод сравнения площадей.

## **1.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

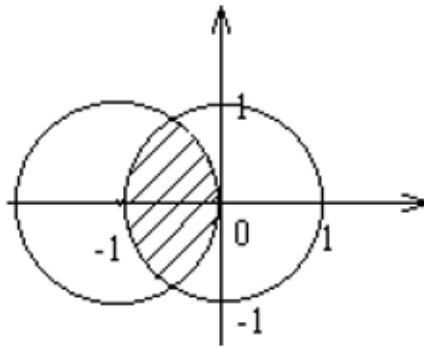
### **Задача**

Создать файл F, состоящий из целых чисел. Переписать из файла F в файл G все четные числа. Вывести на экран содержимое обоих файлов.

## 2 Вариант 2

### 2.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

В заданном целом числе определить количество нечетных цифр.

## 2.2 ЛР 2. Коллекции и строки

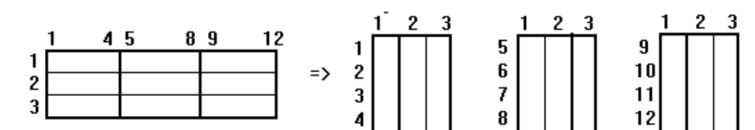
При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Даны координаты 20 точек на прямой. Определить номера двух точек, расстояние между которыми наибольшее (считать, что такая пара точек единственная). Вывести на экран массив, координаты и номера найденных точек.

### Задание 2. Матрицы

Дан массив символов размером 3\*12. Вывести массив на экран в виде трех фрагментов, расположив соответствующие части строк вертикально:



### Задание 3. Строки

Дана непустая последовательность слов из строчных латинских букв (количество слов больше 5): между соседними словами – запятая и пробел, за последним словом – точка. Определить количество слов, которые начинаются на заглавную латинскую букву. Вывести на экран количество слов и сами эти слова.

## 2.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дано четное число  $N > 2$ . Проверить для него гипотезу Гольдбаха, состоящую в том, что каждое четное число представимо в виде суммы двух простых чисел. Вывести на экран само число и простые числа, из которых оно формируется.

## **2.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

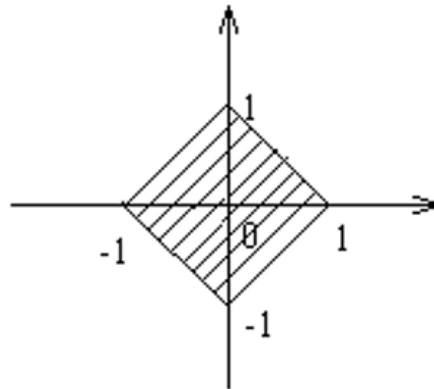
### **Задача**

Создать файл F, состоящий из целых чисел. Переписать из файла F в файл G все нечетные числа. Вывести на экран содержимое обоих файлов.

### 3 Вариант 3

#### 3.1 ЛР 1. Базовое структурное программирование

##### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

##### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

##### Задача

Определить число, получаемое записью в обратном порядке цифр заданного натурального числа. Массивы и строки не использовать.



## 3.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан вещественный массив из 20 элементов. Если за каждым положительным элементом расположены не более двух не положительных, оставить массив без изменений. В противном случае оставить в массиве только отрицательные элементы, сохранив их порядок.

### Задание 2. Матрицы

Дан массив  $A(N,N)$  ( $N \leq 10$ ), каждый элемент которого - символ \*. Вывести массив на экран так, чтобы получилось изображение только одного периметра массива.

### Задание 3. Строки

Дан текст из латинских символов. Группы символов, разделённых одним или несколькими пробелами и не содержащие пробелов внутри себя, назовём словами. Найти количество слов, начинающихся с буквы «В» и заканчивающиеся «К».

### 3.3 ЛР 3. Функциональная и модульная декомпозиция

#### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

#### **Задача**

Дана целочисленная матрица  $A(n, m)$ ,  $(n, m \leq 10)$ . Преобразовать ее, переставив строки в порядке возрастания их наименьших элементов.

### **3.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

#### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

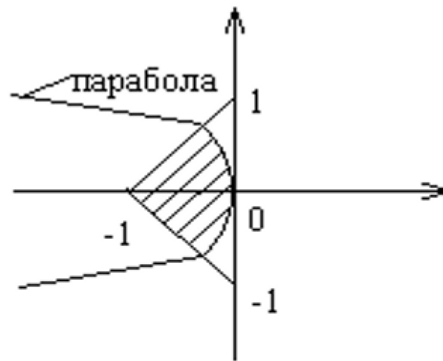
#### **Задача**

Создать текстовый файл F, содержащий слова из латинских символов. Переписать из файла F в файл G все слова, состоящие меньше, чем из трех символов.

## 4 Вариант 4

### 4.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Найти сумму первых  $k$  чисел последовательности Фибоначчи. Последовательность определяется законом:  $F_0 = F_1 = 1; F_n = F_{n-1} + F_{n-2}$  для  $n \geq 2$ .

## 4.2 ЛР 2. Коллекции и строки

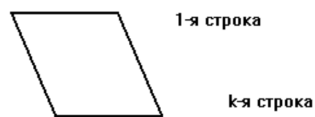
При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Определить число инверсий в массиве из  $n \leq 20$  целых чисел (инверсия – это пара элементов, в которой большее число расположено слева от меньшего).

### Задание 2. Матрицы

Дан трехмерный массив символов  $L(8,6,4)$ . Вывести его элементы на экран строками по 12 символов в виде параллелограмма:



### Задание 3. Строки

Дано натуральное число  $n$ , латинские символы  $S_1, S_2, \dots, S_n$ , образующие строку. Группы символов, разделенных одним или несколькими пробелами и не содержащих пробелов внутри себя, назовем словами. Найти количество слов, начинающихся с буквы «А» и оканчивающиеся на «Е».

### 4.3 ЛР 3. Функциональная и модульная декомпозиция

#### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

#### **Задача**

Дана целочисленная матрица  $A(n, m)$ ,  $(n, m \leq 10)$ . Преобразовать ее, упорядочив каждую строку по неубыванию элементов.

## **4.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

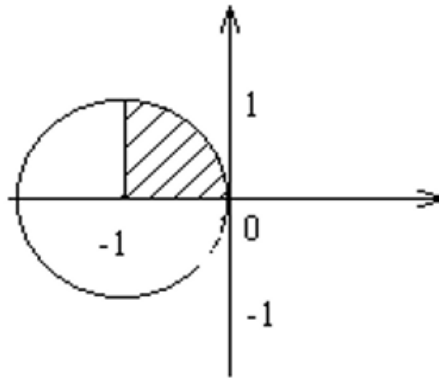
### **Задача**

Создать текстовый файл F, содержащий латинские слова. Переписать из файла F в файл G все слова, состоящие больше чем из трех букв.

## 5 Вариант 5

### 5.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Определить первое число последовательности Фибоначчи, превышающее число  $k$ . Последовательность определяется законом:  $F_0 = F_1 = 1; F_n = F_{n-1} + F_{n-2}$  для  $n \geq 2$ .



## 5.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Даны две последовательности по 30 чисел в каждой. Найти наибольшее среди тех чисел второй последовательности, которые не входят в первую (считать, что хотя бы одно такое число существует).

### Задание 2. Матрицы

Дан массив  $S(6,6)$  целых двухзначных чисел. Вывести элементы массива в виде равнобедренного треугольника следующим образом: строка из 1 элемента, под ней строка из 2 элементов, далее из 3 и т.д. В последней строке — 8 элементов.

### Задание 3. Строки

Дана непустая последовательность слов из строчных латинских букв (количество слов больше 3): между соседними словами — запятая, за последним словом — точка. Определить количество слов, которые содержат более трех гласных букв.

### 5.3 ЛР 3. Функциональная и модульная декомпозиция

#### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

#### **Задача**

Дана квадратная целочисленная матрица  $A$  порядка  $n$  ( $n < 10$ ). В каждой строке поменять местами  $k$ -ый и  $l$ -ый элементы. Напечатать исходную и преобразованную матрицы. Номера  $k$  и  $l$  вводить с клавиатуры в основной программе.

## **5.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

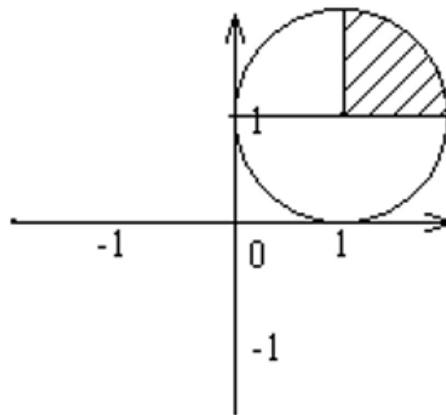
### **Задача**

Создать текстовый файл F. Переписать из файла F в файл G все четные строки.

## 6 Вариант 6

### 6.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Найти все нечетные числа последовательности Фибоначчи, не превышающие заданного числа  $k$ . Последовательность определяется законом:  $F_0 = F_1 = 1$ ;  $F_n = F_{n-1} + F_{n-2}$  для  $n \geq 2$ .

## 6.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан целочисленный массив  $B(n)$ ,  $n < 41$ ,  $n$  четно. Найти наименьший элемент последовательности произведений  $\{b_1 \circ b_n; b_2 \circ b_{n-1}; \dots; b_{n/2} \circ b_{n/2+1}\}$ .

### Задание 2. Матрицы

Дана целочисленная матрица  $A(n,n)$ ,  $n \leq 10$ . Вывести на экран матрицу  $A$ , повернутую на  $90^\circ$ .

### Задание 3. Строки

Дан текст. В конце каждого предложения стоит точка, между словами — ровно один пробел. Определить количество предложений длиннее трех слов.

## 6.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Определить количество простых чисел в целочисленном массиве  $C(n)$ .

## **6.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

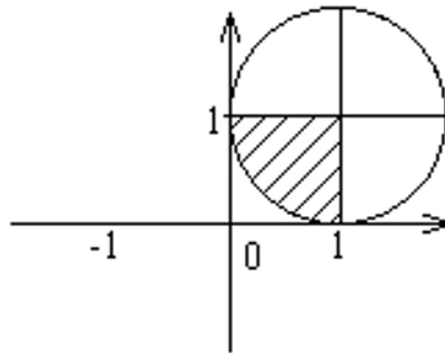
### **Задача**

Создать текстовый файл F. Переписать из файла F в файл G все нечетные строки.

## 7 Вариант 7

### 7.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Вычислить:  $\sqrt{3 + \sqrt{6 + \cdots \sqrt{60 + \sqrt{63}}}}$



## 7.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан целочисленный массив  $N(30)$ . Оставить его без изменения, если он упорядочен по неубыванию. В противном случае удалить из массива все элементы, нарушающие неубывающую последовательность.

Например, если исходный массив: 2, 2, 14, 8, 10, 15, 1, 44, то результатом будет массив: 2, 2, 14, 15, 44.

### Задание 2. Матрицы

Каждым элементом массива  $L(8)$  является символьная строка длиной 15. Вывести на экран элементы массива  $L$  в два столбца: четные строки справа, нечетные слева, причем элементы четных строк должны быть выведены в обратном порядке.

### Задание 3. Строки

Дана последовательность слов латинского алфавита. Длина каждого слова – 5 символов, длина всей последовательности – не более 40 символов. Слова разделены пробелами. Последнее слово заканчивается символом «.». Если слово начинается с букв A, B или C, то третью букву слова заменить на Y, а если слово оканчивается на D, E или F, то предпоследнюю букву заменить на O.

### 7.3 ЛР 3. Функциональная и модульная декомпозиция

#### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

#### **Задача**

Задана матрица  $G(n, n)$ , ( $n \leq 8$ ). В каждой строке поменять местами максимальный и диагональный элементы. Напечатать исходную и преобразованную матрицы. Задана матрица  $G(n, n)$ , ( $n \leq 8$ ). В каждой строке поменять местами максимальный и диагональный элементы. Напечатать исходную и преобразованную матрицы.

## **7.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

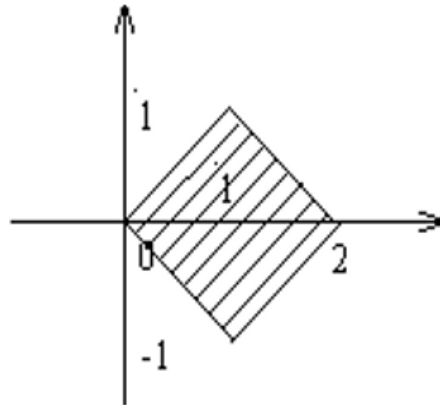
### ***Задача***

Создать текстовый файл F. Удалить из файла F заданную строку.

## 8 Вариант 8

### 8.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Даны два натуральных числа  $N$  и  $M$ . Определить их наибольший общий делитель (НОД), используя алгоритм Евклида.

## 8.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан целочисленный массив  $X(3n)$ . Последовательность элементов этого массива определяет на плоскости  $n$  квадратов со сторонами, параллельными осям координат. Так,  $(x_1, x_2)$  — координаты левого верхнего угла первого квадрата, а  $x_3$  — длина его стороны, и т. д. Определить номера квадратов, которым принадлежит точка с известными координатами  $(p, q)$ . Если таких квадратов нет, то выдать сообщение.

### Задание 2. Матрицы

Массив  $D(m, m)$  заполнить символами \*, \$, и ? произвольным образом (один элемент — один символ). Вывести на экран исходную матрицу и матрицу с отсеченными углами. «Длину»  $k$  катетов отсекаемых треугольников ввести по запросу. Например, для  $m=5$  и  $k=2$ :

*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
	*	*	*	*
*	*	*	*	*

 $\Rightarrow$ 

			*	
		*	*	*
*	*	*	*	*
	*	*	*	
			*	

### Задание 3. Строки

Дана строка, состоящая из 6 слов фиксированной длины по 6 латинских символов. Удалить слова, не содержащие букв «A», «B», «E».

### 8.3 ЛР 3. Функциональная и модульная декомпозиция

#### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

#### **Задача**

Дана вещественная матрица  $A(n, m)$ ,  $(n, m \leq 10)$ . Упорядочить матрицу по невозрастанию сумм элементов ее строк.

## **8.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

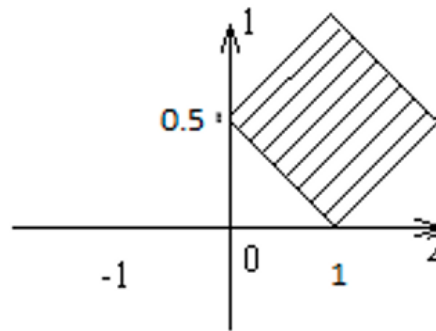
### **Задача**

Создать файл F, содержащий целые числа. Переписать в обратном порядке все числа из файла F в файл G. Вывести на экран содержимое обоих файлов.

## 9 Вариант 9

### 9.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Найти все простые делители данного натурального числа  $N$ .



## 9.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетями). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Ввести целочисленный массив  $A(n)$ ,  $n < 50$ . Удалить числа, встречающиеся в массиве более 3-х раз.

### Задание 2. Матрицы

Заполнить массив  $A(n,n)$ ,  $n \leq 10$ , квадратами первых  $n^2$  натуральных чисел. Вывести его на экран в виде двух частей, «разрезав» по побочной диагонали. Части расположить друг под другом. Побочная диагональ должна присутствовать в обеих частях.

### Задание 3. Строки

Дана строка длиной не более 40 латинских символов. Последний символ — «.». Слова в тексте разделены несколькими пробелами. Определить номер слова, содержащего максимальное количество букв «А».

### 9.3 ЛР 3. Функциональная и модульная декомпозиция

#### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

#### **Задача**

Дана символьная матрица размера  $A(n, m)$ ,  $(n, m \leq 10)$ . Преобразовать ее таким образом, чтобы элементы каждой нечетной строки расположились в обратном порядке. Новую матрицу не заводить.

## 9.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода

### Задание 1. Функции и ввод-вывод

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

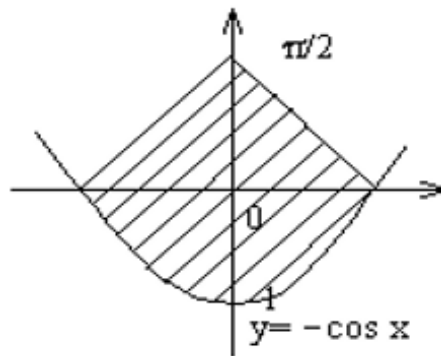
### *Задача*

Создать файл F, содержащий целые числа. Создать новый файл G, состоящий из сумм каждой нечетной компоненты и следующей за ней компоненты из файла F. Вывести на экран содержимое обоих файлов.

## 10 Вариант 10

### 10.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Найти номера двух соседних членов последовательности Фибоначчи, разность между которыми превышает число  $k$ . Последовательность определяется законом:  $F_0 = F_1 = 1; F_n = F_{n-1} + F_{n-2}$  для  $n \geq 2$ . Вывести на экран числа, их номера и разность между числами.

## 10.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Присвоить переменной  $t$  значение «истина», если в массиве  $X$  нет нулевых элементов и, кроме того, положительные и отрицательные элементы чередуются. В противном случае  $t$  должна принять значение «ложь». Вывести на экран значение  $t$ .

### Задание 2. Матрицы

Сформировать массив символов  $Q(11, 11)$ , в первой строке которого расположить латинские буквы от  $a$  до  $k$ ; во второй – те же буквы, но сдвинутые по кольцу на один элемент вправо:  $k, a, b, \dots, i$ , и т.д. Вывести массив на экран так, чтобы буквы  $d$  каждой строки образовывали вертикаль.

### Задание 3. Строки

Дана строка длиной не более 40 символов. Строка разделена точками на подстроки. Поменять местами подстроки минимальной и максимальной длины.

## 10.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Программа идентификации групп в строке. Строка состоит из нескольких групп. Каждая группа представляет собой букву, повторенную некоторое число раз:  
«ccccccccccccdddddffffff.....aaaaaaaaa».

Программа должна формировать новую символьную строку, в которой о каждой группе сообщается ее буква и количество вхождений буквы в группу. Сведения о группах разделяются пробелом. Например: «c14 d11 f13».

## 10.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода

### Задание 1. Функции и ввод-вывод

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

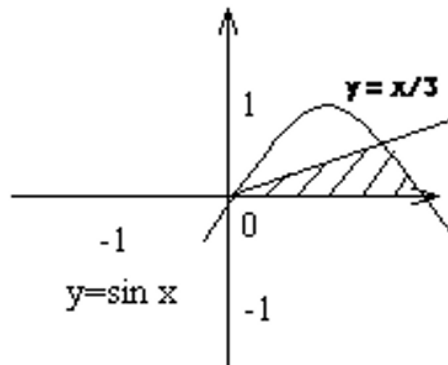
### *Задача*

Создать файл F, содержащий целые числа. Создать новый файл G, состоящий из сумм каждой нечетной компоненты и следующей за ней компоненты из файла F. Вывести на экран содержимое обоих файлов.

## 11 Вариант 11

### 11.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Дано двузначное число  $k$ . Определить  $k$ -ю цифру последовательности 248163264128256... (степени 2). Массивов и строк не использовать.



## 11.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан массив  $Q(30)$ , элементы которого упорядочены по возрастанию. Определить номер элемента, равного числу  $Y$ , и вывести исходный массив и этот номер на экран. Если такого элемента нет, то выдать сообщение. Использовать метод бинарного поиска: сравнить  $Y$  со средним элементом массива, если числа равны, то поиск прекратить, если  $Y$  меньше среднего элемента, то поиск продолжить в первой половине массива, а, если  $Y$  больше, то поиск продолжить во второй половине массива. К выбранной половине применяется тот же алгоритм.

### Задание 2. Матрицы

Дан целочисленный массив  $P(6,3)$ . Вывести его на экран по строкам следующим образом: нечетные строки горизонтально одна под другой; четные — рядом вертикально, одна за другой.

1	—	—	—	2	4	6
3	—	—	—			
5	—	—	—			

### Задание 3. Строки

Дана строка длиной не более 40 символов латинского алфавита, состоящая из последовательности трехсимвольных слов, разделённых пробелами. У всех слов, начинающихся с букв M,N,K, второй символ заменить на E, а у всех слов, оканчивающихся на S, первый символ заменить на Z.

## 11.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Даны три матрицы разных порядков. Найти сумму их наименьших элементов (считая, что в каждой матрице такой элемент единственный).

## **11.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

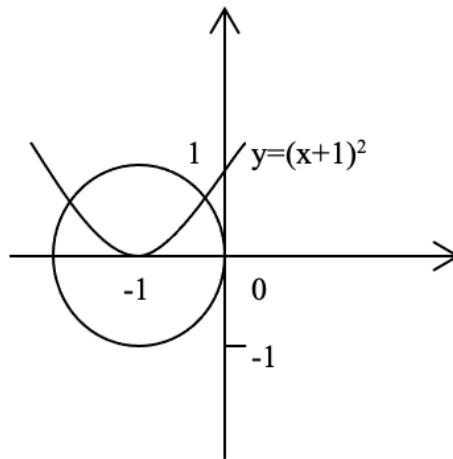
### **Задача**

Создать файл F, содержащий целые числа. Переписать в файл G в обратном порядке все четные числа.

## 12 Вариант 12

### 12.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Присвоить переменной  $P$  значение «истина» в том случае, если среди чисел  $\sin x^n$  ( $n = 1, 2, 3, \dots, k$ ) есть хотя бы одно отрицательное, и «ложь» – в противном случае. Вывести на экран значение  $P$ . Проверить программу при различных  $k$ , которые вводятся с клавиатуры.

## 12.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

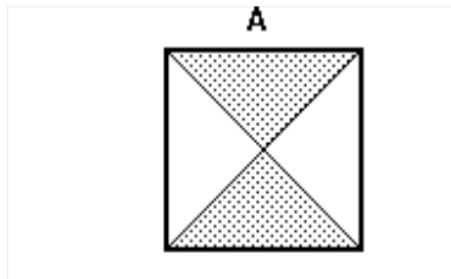
Преобразовать массив  $S(60)$  так, чтобы получилась последовательность вида:

$\{s_1, s_{21}, s_{22}, s_2, s_{23}, s_{24}, s_3, \dots, s_{59}, s_{60}\}$

Вывести на экран исходный и преобразованный массив.

### Задание 2. Матрицы

Дан целочисленный массив  $A(11,11)$ . Вывести на экран заштрихованную часть массива  $A$ , включая диагональные элементы. Исходный массив не изменять!



### Задание 3. Строки

Дана строка длиной не более 40 символов. Слова в строке разделены одним пробелом. Последний символ — «.». Удалить из строки слова длиной не более 3-х символов.

## 12.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дана строка из  $n$  символов. Для каждого символа строки определить его категорию: 1 — латинские буквы, 2 — цифры, 3 — прочие символы. Вывести на экран каждый символ и номер его категории.

## 12.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода

### Задание 1. Функции и ввод-вывод

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

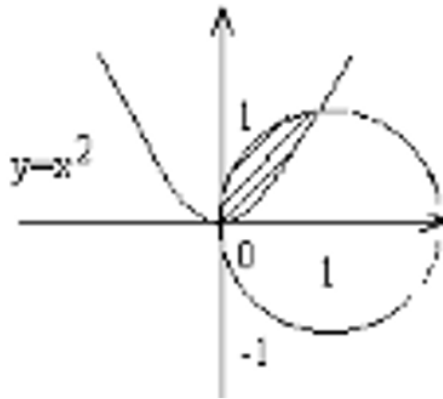
### *Задача*

Задан файл F, состоящий из записей. Каждая запись содержит название города и год его основания. Переписать в файл G сведения обо всех городах, название которых начинается на букву «А». Вывести на экран содержимое обоих файлов.

## 13 Вариант 13

### 13.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Определить наибольшее число последовательности  $f = e^{\sqrt{2n-1}}$ , не превышающее заданного числа  $k$  ( $n = 1, 2, 3 \dots$ ). Число  $k$  вводится с клавиатуры.



## 13.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Написать программу обработки одномерного массива. Дан целочисленный массив  $A(30)$ . Получить наибольший элемент последовательности  $\{a_1 + a_{30}; a_2 + a_{29}; \dots; a_{15} + a_{16}\}$ .

### Задание 2. Матрицы

Целочисленный массив размера  $n \times m$  ( $n, m \leq 10$ ) сформировать следующим образом: все элементы первого столбца равны 10, второго - 20, третьего - 30 и т. д. Вывести массив на экран так, чтобы каждая четная строка была смещена относительно нечетной на два элемента поочередно то вправо, то влево:

```

                * * * * *
            * * * * *
        * * * * *
            * * * * *
                * * * * *
            * * * * *
        * * * * *
```

... и т.д.

### Задание 3. Строки

Дано натуральное число  $n$ , символы  $S_1, S_2, \dots, S_n$ , образующие строку. Группу символов, разделенных одним или несколькими пробелами и не содержащих пробелов внутри себя, назовем словами. Найти количество слов, у которых первый и последний символ совпадают.

### 13.3 ЛР 3. Функциональная и модульная декомпозиция

#### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

#### **Задача**

В строке из  $n$  произвольных символов заменить все строчные латинские буквы на прописные, а прописные на строчные.

## **13.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

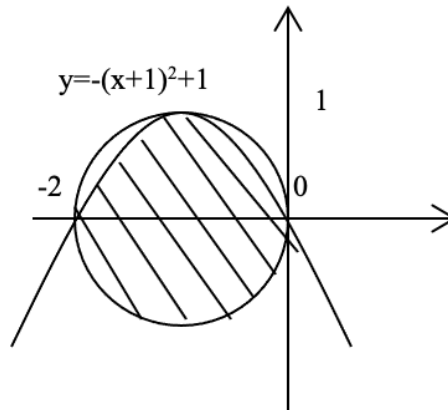
### **Задача**

Создать текстовый файл F. Переписать в файл G в обратном порядке все строки файла F.

## 14 Вариант 14

### 14.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Задано натуральное число  $N$ . Определить, является ли оно простым.

## 14.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетями). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Переписать элементы целочисленного массива  $C(30)$  в обратном порядке (новый массив не использовать). В полученном массиве найти сумму максимального и минимального элементов. Вывести на экран исходный и полученный массивы, найденные элементы и полученную сумму.

### Задание 2. Матрицы

Сформировать массив  $D(n,n)$ ,  $n \leq 10$ , следующим образом: элемент массива равен 1, если он расположен на диагонали (главной или побочной), в противном случае он равен 0. Вывести на экран отдельно правую и левую половины массива так, чтобы они были расположены рядом, но правая половина была бы опущена на 1 строку вниз по отношению к левой.

### Задание 3. Строки

Дана строка длиной не более 80 латинских символов. Символы образуют слова, отделенные друг от друга запятыми, последний символ – точка. Определить количество слов в последовательно-сти, оканчивающихся на букву «a». Вывести на экран исходную строку, найденные слова и их количество. Если таких слов нет, напечатать соответствующее сообщение.

## 14.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Вводится натуральное число  $N$ . Среди чисел  $1, 2, \dots, N$  найти все те, которые можно представить в виде суммы квадратов двух натуральных чисел. Вывести на экран найденные числа и числа, из квадратов которых они составлены.

## **14.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

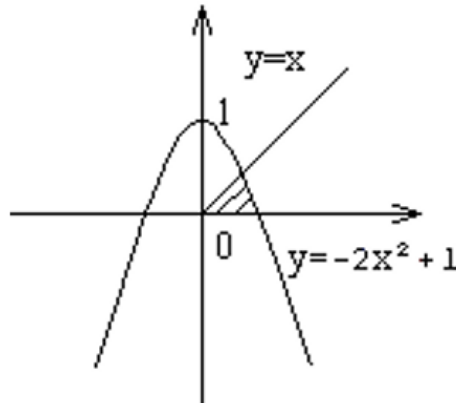
### **Задача**

Создать текстовый файл F. Удалить из файла все строки с нечетными номерами.

## 15 Вариант 15

### 15.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Найти все натуральные числа из диапазона  $[111, 999]$ , состоящие только из нечетных цифр.



## 15.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

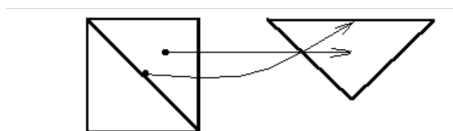
В результате измерений получено  $n \leq 50$  значений некоторой целой величины  $B$ , которые вводятся с клавиатуры. Написать и отладить программу определения суммы элементов, кратных 3, и количества отрицательных элементов последовательности. Вывести на экран исходный массив и результаты с комментариями.

Исходные данные для отладки:

$n = 10, B = \{12839, -81, 54, -33, 82, 9, 18, 51\}$

### Задание 2. Матрицы

Символьный массив размером  $p \times p$  элементов ( $p \leq 10$ ) «разрезать» по главной диагонали. Вывести на экран ту часть массива, которая расположена выше главной диагонали (включая ее), развернув ее так, как показано на рисунке. Символы выводить через пробел.



### Задание 3. Строки

Дана строка длиной не более 80 латинских символов. Символы образуют слова, отделенные друг от друга запятыми, последний символ — точка. Определить количество четных слов в последовательности, начинающихся с буквы «t». Вывести на печать исходную строку, найденные слова и их количество. Если таких слов нет, напечатать соответствующее сообщение.

## 15.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дана последовательность из  $4n$  целых чисел,  $n \leq 10$ . В каждой из  $n$  четверок первая пара чисел — координаты ферзя на шахматной доске, вторая пара — координаты пешки. Каждая координата может иметь значение от 1 до 8. Определить, в скольких из  $n$  случаев расположения фигур ферзь бьет пешку.

*Примечание:* Ферзь бьет фигуры по вертикали, горизонтали и диагонали на любом расстоянии.

## **15.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

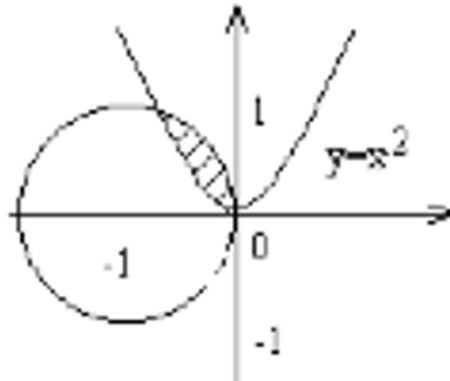
### ***Задача***

Дан файл F, состоящий из записей. Каждая запись содержит название страны, название города и численность его населения. Переписать из файла F в файл G сведения обо всех городах заданной страны (страна вводится с клавиатуры). Вывести на экран содержимое обоих файлов.

## 16 Вариант 16

### 16.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Задано натуральное число  $n$  ( $n > 7$ ). Доказать, что существует пара натуральных чисел  $(x, y)$  таких, что  $5x + 3y = n$ , где  $x$  принимает минимально возможное значение.

## 16.2 ЛР 2. Коллекции и строки

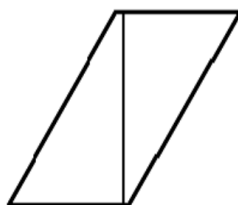
При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан целочисленный массив  $A(n)$  ( $n \leq 34$ ). Исключить из него все элементы, равные максимальному и минимальному, сохранив порядок остальных. Вывести на экран исходный и полученный массивы.

### Задание 2. Матрицы

Сформировать массив размерностью  $n \times m$  ( $n, m \leq 10$ ) следующим образом: нечетные строки содержат целые числа от 1 до  $m$ , а четные — их квадраты. Вывести массив на экран в виде параллелограмма так, чтобы главная диагональ располагалась вертикально.



### Задание 3. Строки

Дано натуральное число  $n$ , символы  $S_1, S_2, \dots, S_n$ . Группы символов, разделенных одним или несколькими пробелами и не содержащие пробелов внутри себя, назовем словами. Подсчитать количество слов в данной последовательности и определить длину наименьшего из них. Вывести на экран исходную строку и найденное слово.

## 16.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дан массив из  $n$  натуральных чисел. Определить количество чисел, в десятичной записи которых используется цифра 7.

## **16.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

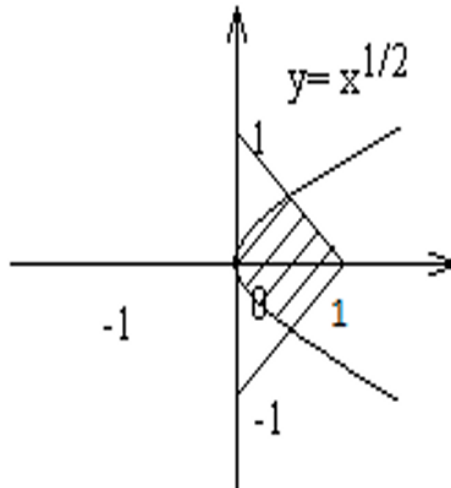
### **Задача**

Создать текстовый файл F. Вставить заданное предложение после строки с номером k.

## 17 Вариант 17

### 17.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Найти сумму цифр заданного натурального числа.



## 17.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетями). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Найти наименьший элемент массива  $V(n)$  ( $n \leq 50$ ). Если он имеет четный номер, поменять местами найденный элемент с первым элементом массива, если нечетный – то с последним. Вывести на экран исходный и полученный массивы.

### Задание 2. Матрицы

Дан массив  $C(7)$ , каждый элемент которого – строка длиной 10 символов. Вывести массив на экран двумя способами:

1. строки расположить вертикально рядом друг с другом;
2. каждую строку, начиная со второй, сдвинуть вправо на один элемент по отношению к предыдущей.

### Задание 3. Строки

Дано натуральное число  $n$ , латинские символы  $S_1, S_2, \dots, S_n$ . Группы символов, разделенных одним или несколькими пробелами и не содержащие пробелов внутри себя, назовем словами. Найти количество слов, начинающихся с буквы «В» и оканчивающихся на «С».

## 17.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### Задача

Даны  $a, b, c$  — длины сторон треугольника. Найти длины медиан треугольника, сторонами которого являются медианы треугольника со сторонами  $a, b, c$ .

*Примечание:* Длина медианы, проведенной к стороне  $x$  треугольника со сторонами  $x, y$  и  $z$ , равна:  
 $0,5\sqrt{2y^2 - x^2 + 2z^2}$

## **17.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

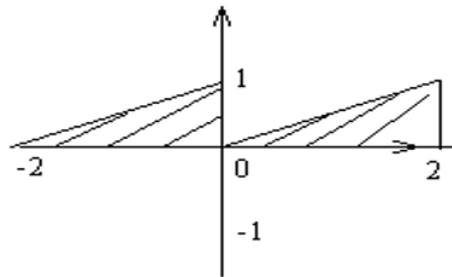
### **Задача**

Даны два файла F и G, содержащие целые числа. Переписать в файл H все числа, входящие в F и не входящие в G. Вывести на экран содержимое всех файлов.

## 18 Вариант 18

### 18.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Найти произведение цифр заданного натурального числа.

## 18.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Элементами массива  $f(40)$  являются цифры. Определить, какая из них встречается чаще других. Вывести на экран исходный массив, найденную цифру и количество ее повторений.

### Задание 2. Матрицы

Дан массив размером  $9 \times 9$  символов. Вывести на экран главную и побочную диагонали массива так, чтобы побочная диагональ была расположена вертикально, а главная по отношению к ней не изменила своего положения.

### Задание 3. Строки

Дана непустая последовательность слов из строчных латинских букв (количество слов больше 3): между соседними словами — запятая, за последним словом — точка. Определить количество слов с четными номерами, которые имеют длину более трех букв. Вывести строку и найденные слова. Если таковых нет – вывести сообщение.

## 18.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Даны  $n$  натуральных чисел — длины отрезков. Для каждой тройки этих отрезков, из которых можно построить треугольник, вывести площадь треугольника. Если таковых нет — вывести соответствующее сообщение.

## **18.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

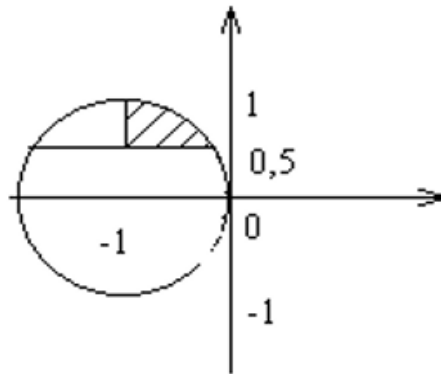
### **Задача**

Задан файл F, состоящий из записей. Каждая запись содержит название книги, автора и год издания. Переписать в файл G все записи, содержащие сведения о книгах, изданных в заданном году. Вывести на экран содержимое обоих файлов.

## 19 Вариант 19

### 19.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Определить количество чисел последовательности  $(n-1)^2/n$ ,  $(n = 1, 2, 3, \dots, \infty)$  попадающих в интервал  $[h, m]$ , где  $h, m$  - вещественные числа ( $m > h$ ), вводимые с клавиатуры. Вывести на экран найденные числа и их количество.



## 19.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан целочисленный массив  $X(4n)$ . Последовательность элементов этого массива определяет на плоскости  $n$  прямоугольников со сторонами, параллельными осям координат. Так,  $(x_1, x_2)$  – координаты левого верхнего угла первого прямоугольника,  $x_3$  – длина его горизонтальной стороны, а  $x_4$  – длина его вертикальной стороны, и т. д. Определить номера прямоугольников, которым принадлежит точка с известными координатами  $(p, q)$ . Если таких прямоугольников нет, то выдать сообщение.

### Задание 2. Матрицы

Дан трехмерный массив  $R(10, 10, 2)$ . Каждым элементом первого квадрата  $10 \times 10$  является символ «!», каждым элементом второго – символ «?». Вывести массив на экран в виде прямоугольника так, чтобы столбец восклицательных знаков чередовался со столбцом вопросительных.

### Задание 3. Строки

Дана строка длиной не более 40 латинских символов. Последний символ – «!». Строка разделена точками на подстроки. Определить количество и номера подстрок, содержащих более трех символов «А». Вывести на экран исходную строку, найденные подстроки и их количество. Если таких подстрок нет – вывести сообщение.

## 19.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дана матрица  $(n, m)$ ,  $(n, m \leq 20)$ . Определить суммы элементов тех строк матрицы, максимальные элементы которых не превышают среднего значения элементов матрицы.

## **19.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

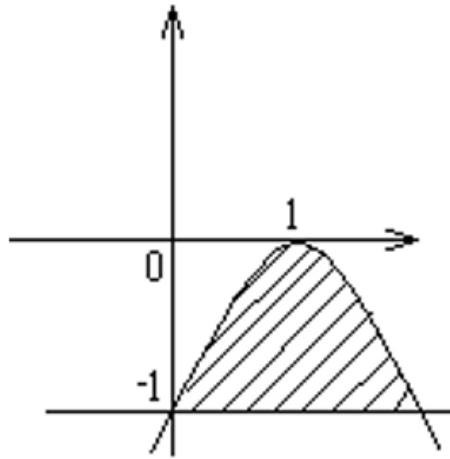
### **Задача**

Задан файл F, состоящий из записей. Каждая запись содержит название книги, автора и год издания. Удалить из файла все записи, содержащие сведения о книгах заданного автора. Вывести на экран содержимое файла до и после удаления.

## 20 Вариант 20

### 20.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Вычислить:  $\sqrt{5 + \sqrt{10 + \sqrt{15 + \dots \sqrt{60}}}}$

## **20.2 ЛР 2. Коллекции и строки**

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### **Задание 1. Одномерные массивы**

Дан одномерный массив чисел. Определить максимальное, минимальное и среднее арифметическое чисел, превышающих заданное пользователем значение  $d$ . Вывести на экран исходный массив и найденные значения со всеми комментариями. Если таковых не было обнаружено – выдать соответствующее сообщение.

### **Задание 2. Матрицы**

Дан целочисленный массив размером  $7 \times 7$ , каждый элемент которого – цифра от 0 до 9. Вывести массив на экран по строкам, сначала все четные строки, затем нечетные.

### **Задание 3. Строки**

Дана строка длиной не более 100 символов, содержащая последовательность команд, разделенных запятыми. Команды: up, down, left, right. Определить, сколько раз каждая команда встречается в строке.

## 20.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Четырехугольник задан координатами своих вершин. Кроме того, даны координаты  $m$  точек на плоскости. Определить, сколько точек находится внутри фигуры. Для определения принадлежности точки многоугольнику использовать метод трассировки луча. Пояснить, в каких случаях метод дает неверный результат и почему.

## **20.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

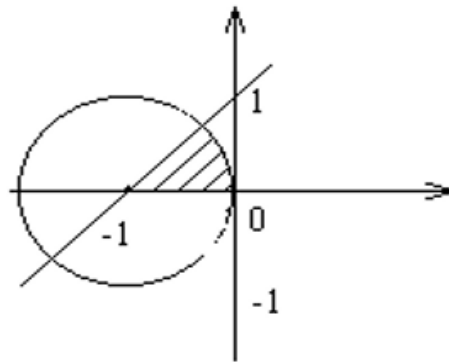
### **Задача**

Задан файл F, состоящий из записей. Каждая запись содержит название книги, автора и год издания. Удалить из файла все записи о книгах, которые изданы в заданном году. Вывести на экран содержимое файла до и после удаления.

## 21 Вариант 21

### 21.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Определить порядковый номер минимального положительного числа последовательности, заданной рекуррентным соотношением:  $x_1 = x_2 = -9; x_n = x_{n-1} + x_{n-2} + 12$ . Вывести на экран как само число, так и его номер.



## 21.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Даны координаты 10 точек на прямой. Определить номера двух точек, расстояние между которыми наименьшее (считать, что такая пара точек единственная). Вывести на экран массив, координаты и номера найденных точек.

### Задание 2. Матрицы

Дана символьная матрица  $M(n, n), n \leq 10$ . Вывести на экран исходную матрицу, а затем эту же матрицу, повернув ее на  $180^\circ$ .

### Задание 3. Строки

Дана строка длиной не более 40 латинских символов. Последний символ – «.». Слова в тексте разделены несколькими пробелами. Удалить слово максимальной длины. Вывести на экран исходную строку, найденное слово, его длину и скорректированную строку.

## 21.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Даны координаты вершин выпуклого четырехугольника и координаты  $m$  точек на плоскости. Определить, сколько точек находится внутри фигуры. Для определения принадлежности точки многоугольнику использовать метод сравнения площадей.

## **21.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

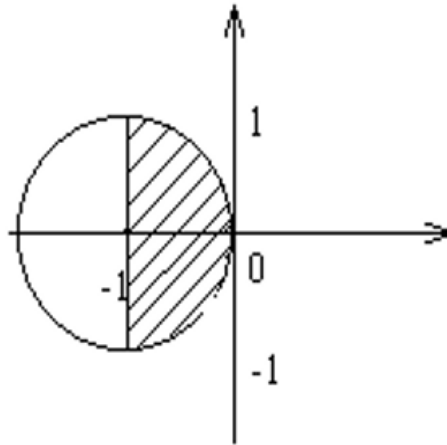
### **Задача**

Задан типизированный файл F, состоящий из записей. Каждая запись содержит название книги, автора и год издания. Переписать в текстовый файл G названия всех книг в обратном порядке. Вывести на экран содержимое обоих файлов.

## 22 Вариант 22

### 22.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Дано натуральное число  $n$ . Выбросить из записи этого числа цифры 1 и 5, оставив порядок остальных без изменения. Например, из числа 59015518 должно получиться число 908. Вывести на печать исходное и полученное числа. Массивов и строк не использовать.

## 22.2 ЛР 2. Коллекции и строки

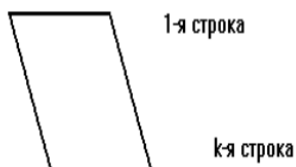
При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан массив из 40 целых чисел. Определить минимальный элемент и количество элементов, равных ему. Вывести на экран исходный массив, найденное минимальное значение и количество чисел, равных ему. Если таких чисел больше нет – выдать сообщение.

### Задание 2. Матрицы

Дан массив символов  $L(8,6)$ . Вывести его на экран строками по 12 символов в виде параллелограмма, где каждая следующая строка смещена относительно предыдущей на одну позицию.



### Задание 3. Строки

Дана строка длиной не более 40 латинских символов. Последний символ – «.». Удалить из строки слова длиной 1 символ. Слова в строке разделены несколькими пробелами. Вывести на экран исходную и полученную строки.

## 22.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Даны координаты вершин треугольника и координаты  $m$  точек на плоскости. Определить, сколько точек находится внутри фигуры. Для определения принадлежности точки треугольнику использовать метод трассировки луча. Пояснить, в каких случаях метод дает неверный результат и почему.

## **22.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

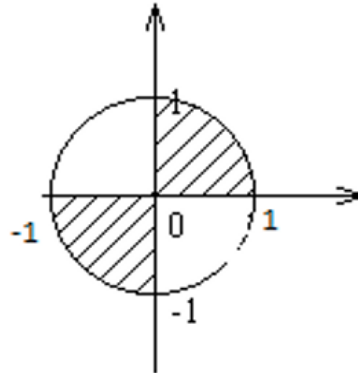
### **Задача**

Создать файл F, состоящий из целых чисел. Переписать из файла F в файл G все числа, кратные трем. Вывести на экран содержимое обоих файлов.

## 23 Вариант 23

### 23.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Определить количество цифр, кратных трем, в записи целого неотрицательного числа. Массивов и строк не использовать.



## 23.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан вещественный массив из 45 элементов. Преобразовать массив следующим образом: сначала расположить все положительные числа, затем все отрицательные и в конце – нули. Вывести на экран исходный и сформированный массив. Вспомогательного массива не использовать.

### Задание 2. Матрицы

Решить поставленную задачу, используя средства управления вводом/выводом. Дан массив размером  $N \times N$  ( $N \leq 10$ ), каждый элемент которого – символ \*. Вывести сначала исходный массив, а затем вывести только главную и побочную диагонали массива.

### Задание 3. Строки

Дана строка S длиной до 40 символов, содержащая разделенные запятыми слова. Дана вторая строка F длиной до 5 символов. Найти в строке S все слова, в которых встречается подстрока F. Вывести на экран исходные строки и результаты поиска.

## 23.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дана целочисленная матрица размера  $B(n, m)$ ,  $(n, m \leq 10)$ . Преобразовать ее, заменив в каждой строке диагональный элемент на максимальный элемент этой строки.

## **23.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

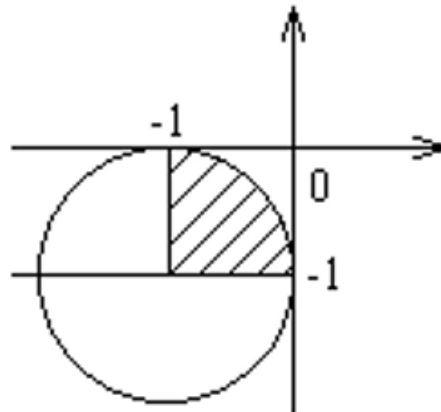
### **Задача**

Задан файл F, состоящий из записей. Каждая запись содержит название города и год его основания. Определить самый старый город и переписать в файл G сведения обо всех городах, основанных в том же году. Вывести на экран содержимое обоих файлов.

## 24 Вариант 24

### 24.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Дано натуральное число  $k$ . Определить  $k$ -ю цифру последовательности 1491625364964... (квадраты натуральных чисел). Массивов и строк не использовать.

## 24.2 ЛР 2. Коллекции и строки

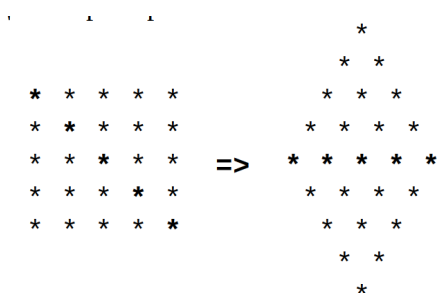
При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

## Задание 1. Одномерные массивы

Даны две последовательности по 15 чисел в каждой. Найти среднее арифметическое тех чисел первой последовательности, которые не входят во вторую (считать, что хотя бы одно такое число существует). Вывести на экран оба исходных массива, найденные числа, и их среднее.

## Задание 2. Матрицы

Дан массив S(5,5) целых однозначных чисел. Вывести сначала исходный массив, а затем элементы массива в виде ромба, расположив главную диагональ горизонтально. Числа выводить через пробел.



### Задание 3. Строки

Дана строка не более 40 латинских символов. Слова разделены несколькими пробелами. Удалить 3 и 5 слова, если они не содержат буквы «А». Вывести на экран исходную строку и ее же после корректировки.

## 24.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дана квадратная целочисленная матрица  $A$  порядка  $n, n \leq 7$ . Выделяя на главной диагонали последовательно по одному элементу  $A_{ii}$ , можно получить  $n$  матриц, ограниченных элементами  $A_{11}$  и  $A_{ii}$ . Программа должна сформировать вектор, элементами которого являются наибольшие элементы всех таких матриц.

## **24.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

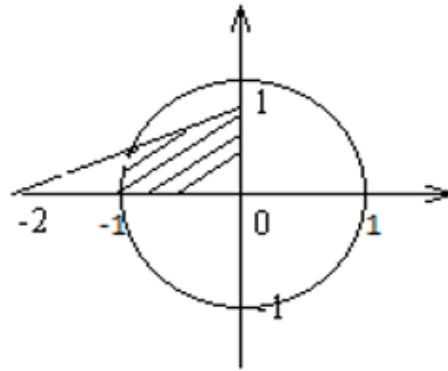
### **Задача**

Создать файл F, состоящий из четного количества целых чисел. Переписать из файла F в файл G суммы первого и второго числа, третьего и четвертого, и т.д. Вывести на экран содержимое обоих файлов.

## 25 Вариант 25

### 25.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Заданы два целых числа  $N$  и  $M$ , определить являются ли они взаимно простыми. Целые числа называются взаимно простыми, если они не имеют никаких общих делителей, кроме  $\pm 1$ . Пример: 14 и 25 взаимно просты.



## 25.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетями). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан массив  $T(3n)$ , состоящий из  $n$  троек целых чисел. Для каждой тройки проверить, можно ли построить треугольник с такими сторонами. Вывести на экран исходный массив и номера троек, из которых можно построить треугольники.

### Задание 2. Матрицы

Дан целочисленный массив  $A(n,n)$  ( $n \leq 6$ ). Вывести его на экран, рядом вывести этот же массив, но вместо правой его части – вертикально отраженную левую часть. Элементы массива не изменять.

1	2	3		1	2	1
4	5	6	=>	4	5	4
7	8	9		7	8	7

### Задание 3. Строки

Дана строка длиной не более 40 латинских символов. Последний символ – «.». Слова в тексте разделены несколькими пробелами. Определить номер слова, в котором более половины символов – буквы «A». Вывести на экран исходную строку, найденное слово и его номер. Если такого слова нет – вывести сообщение.

## 25.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дан целочисленный массив  $T(3n)$ ,  $n \leq 10$ , в котором хранятся длины сторон  $n$  треугольников. Первые три элемента массива — стороны первого треугольника и т.д. Для каждого треугольника определить и вывести на экран его тип: остроугольный, прямоугольный или тупоугольный.

## **25.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

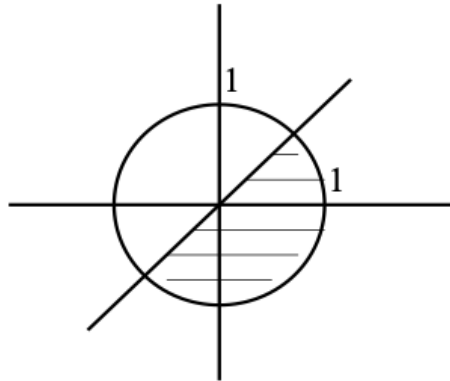
### **Задача**

Создать текстовый файл F. Переписать из файла F в файл G все слова, в которых первая и последняя буквы совпадают.

## 26 Вариант 26

### 26.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Последовательность задана следующим образом:  $a_1 = 12, a_2 = 10, a_n = a_{n-1} + a_{n-2} - 2$ , для  $n \geq 3$ . Найти наибольший член последовательности, меньший 100. Вывести найденный член последовательности и его номер.

## 26.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетями). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан целочисленный массив  $N(30)$ . Определить наибольшее число в массиве. Удалить из массива все элементы, равные наибольшему. Вывести на экран исходный и полученный массивы, а также наибольшее число.

### Задание 2. Матрицы

Дан символьный массив  $S(n,n)$ ,  $n \leq 10$ . Вывести сначала исходный массив, а затем элементы массива в виде ромба, расположив главную диагональ вертикально. Элементы массива выводить через пробел.

```

      *
     * *
    * * *
   * * * *
  * * * * *
 =>
      *
     * *
    * * *
   * * * *
  * * * * *
     * *
      *
```

### Задание 3. Строки

Дана строка, состоящая 6 слов фиксированной длины по 6 латинских символов. Удалить слова, не содержащие букв «A», «D», «E». Вывести на экран исходную строку и полученный результат. Если таких слов нет – выдать сообщение.

## 26.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дана последовательность из  $m$  натуральных чисел,  $m \leq 20$ . Определить, сколько чисел являются палиндромами (одинаково читаются слева направо и справа налево).

## **26.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

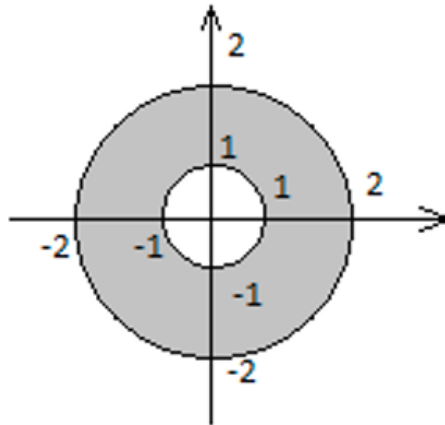
### **Задача**

Создать текстовый файл F. Переписать из файла F в файл G все слова, начинающиеся с заглавной латинской буквы.

## 27 Вариант 27

### 27.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Задано натуральное число  $N$ . Определить, является ли оно совершенным. Число называется совершенным, если равно сумме всех своих делителей, меньших  $N$  (например, число 6 совершенно:  $6 = 1 + 2 + 3$ ).



## 27.2 ЛР 2. Коллекции и строки

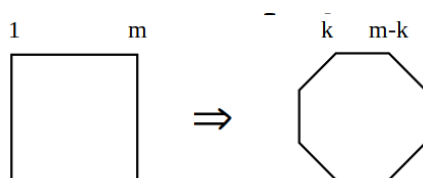
При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан целочисленный массив  $S(n)$  ( $n \leq 50$ ). Определить количество элементов, кратных 2, сумму элементов, кратных 3, и среднее арифметическое отрицательных элементов. Вывести на экран исходный массив и найденные величины или соответствующее сообщение об отсутствии указанных элементов.

### Задание 2. Матрицы

Матрицу  $D(m, m)$  ( $m \leq 12$ ) заполнить символами \*, \$, и ? произвольным образом (один элемент – один символ). Вывести на экран исходную матрицу и часть матрицы так, как показано на рисунке. Число  $k \leq \lfloor m/2 \rfloor$  ввести по запросу.



### Задание 3. Строки

Дана строка длиной до 100 символов, содержащая разделенные точкой предложения. Определить количество предложений, начинающихся с заглавной латинской буквы и оканчивающихся на букву «t».

## 27.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Дана последовательность символьных слов, разделенных пробелом. Для каждого слова определить, является ли оно палиндромом (одинаково читается слева направо и справа налево).

## **27.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

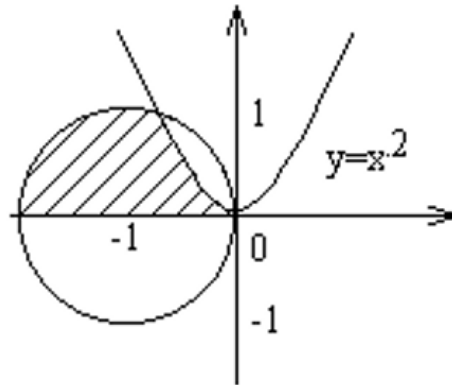
### **Задача**

Создать текстовый файл F. Переписать из файла F в файл G все строки, усекая их до 10 символов.

## 28 Вариант 28

### 28.1 ЛР 1. Базовое структурное программирование

#### Задание 1



Ввести два вещественных числа  $X$  и  $Y$ . Определить, принадлежит ли точка с координатами  $(X, Y)$  заштрихованной части плоскости. Протестировать все ветви алгоритма.

#### Задание 2

Рассмотреть решение предложенной задачи с использованием двух видов циклов:

- `while ... end`
- `for ... end`

Реализовать и отладить программу с наиболее рациональным вариантом цикла. Обосновать выбор.

#### Задача

Последовательность  $x_1, x_2, \dots, x_n$  образована по закону:

$$x_1 = 0; x_2 = 5/8; x_i = (x_{i-1})/2 + (3/4) \times x_{i-2}; (i = 3, 4, \dots).$$

Дано натуральное  $n > 3$ . Вычислить член последовательности с номером  $n$ . Номер  $n$  вводится с клавиатуры.

## 28.2 ЛР 2. Коллекции и строки

При выполнении этой работы для обработки данных можно пользоваться любыми коллекциями Julia из лекционного материала (массивами, словарями, сетами). Необходимо использовать функции.

### Задание 1. Одномерные массивы

Дан целочисленный массив  $A(n)$  ( $n \leq 45$ ). Исключить из него все элементы, большие среднего арифметического всех элементов, сохранив порядок остальных. Вывести на экран исходный и полученный массивы и значение среднего арифметического.

### Задание 2. Матрицы

Сформировать массив символов размерностью  $8 \times 8$  элементов. Вывести массив целиком, а также его представление в виде шахматной доски: показать символы, стоящие на месте черных клеток, и скрыть символы, стоящие на месте белых клеток. Элементы массива не изменять.

```
* * * *      *      *
* * * * => *      *
* * * *      *      *
* * * *      *      *
```

### Задание 3. Строки

Дана символьная строка, состоящая из латинских слов, разделенных запятыми. Подсчитать количество слов в данной последовательности и определить длину наименьшего и наибольшего из них. Вывести на экран исходную строку, найденные слова и их длину.

## 28.3 ЛР 3. Функциональная и модульная декомпозиция

### Задание 1. Функции и модули

Решить задачу, используя функции и модули. На примере полученной программы продемонстрировать умение:

1. Помещать весь код программы в функции.
2. Помещать выполняющие служебные действия функции в отдельный модуль (или несколько, если необходимо).
3. Помещать модули в отдельные файлы.
4. Открывать и импортировать требуемый код из модулей.

Дополнительным плюсом в работе являются:

1. Отделение вычисляющих (чистых) функций от функций ввода-вывода.
2. Юнит-тесты написанных функций.

### **Задача**

Даны координаты начала  $(x_1, y_1)$  и конца  $(x_2, y_2)$  отрезка, а также координаты  $n$  точек на плоскости. Определить, сколько из заданных точек лежит на отрезке.

Примечание. Точка с координатами  $(x, y)$  лежит на отрезке, если существует число  $t$  такое, что выполняются условия:  $x = (1 - t)x_1 + tx_2, y = (1 - t)y_1 + ty_2$  и  $0 \leq t \leq 1$ .

## **28.4 ЛР 4. Пользовательские структурные типы и операции ввода-вывода**

### **Задание 1. Функции и ввод-вывод**

Решить задачу, используя функциональную декомпозицию кода, самостоятельно разработанные структурные типы и рассмотренные на лекции приёмы корректной работы с файловым вводом-выводом.

В работе необходимо показать умение:

1. Отделения вычисляющих (чистых) функций от функций ввода-вывода.
2. Использование структурных типов для промежуточного хранения результатов вычислений и передачи их между функциями вычисления и функциями ввода-вывода.
3. Написания юнит-тестов написанных функций.

### **Задача**

Создать текстовый файл. Заменить в нем строку с номером  $k$  на введенное с клавиатуры предложение.