



Лабораторная 05. Функции

Обязательная часть. Сделать, загрузить на Ваш GitHub, на ТУИС РУДН прикрепить GitHub ссылку:

Упражнение 1.

Напишите программу, отвечающую пользователю на вопрос, принадлежит ли точка с координатами (x, y) закрашенной области.

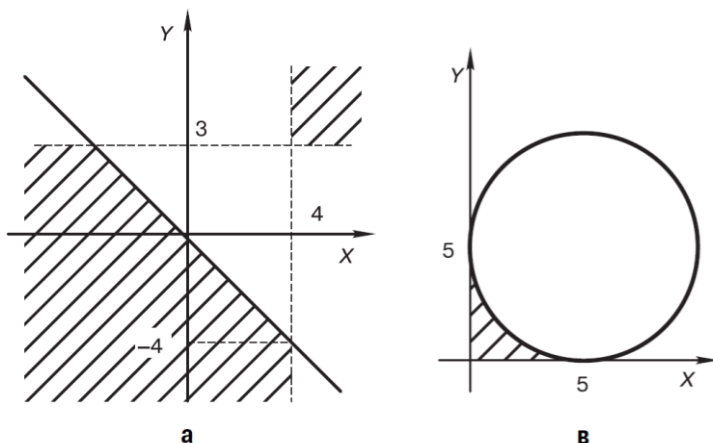


График а: доделать классную работу ТОЛЬКО с использованием логических выражений (без `if...else` или `a > b ? результат1 : результат2`), а именно:

- 1) Исправить логические ошибки (например, вывод для координат $(-5, 4)$ некорректен)
- 2) Написать 10 тестов (10 inputs of (x, y)) и обосновать в комментарии рядом, какое сомнение Вы хотели развеять проверкой данной проверкой.
- 3) изменить, где необходимо тип `int` на `Boolean` (см. [Статью](#) – обратите внимание на стиль названий переменных данного типа)
- 4) дать возможность пользователю самому ввести через терминал желаемые координаты

График в: сделать С ИСПОЛЬЗОВАНИЕМ `if...else` или `a > b ? результат1 : результат2` и не забыть:

Написать 10 тестов (10 inputs of (x, y)) и обосновать в комментарии рядом, какое сомнение Вы хотели развеять данной проверкой.

Упражнение 2.

Напишите программу, в которой определены следующие четыре функции:

1. Функция `getInput()` не имеет параметров, запрашивает ввод с клавиатуры и возвращает в основную программу полученную строку.
2. Функция `testInput()` имеет один параметр. В теле она проверяет, можно ли переданное ей значение преобразовать к целому числу. Если можно, возвращает логическое `True`. Если нельзя – `False`.
3. Функция `strToInt()` имеет один параметр. В теле преобразовывает переданное значение к целочисленному типу. Возвращает полученное число.



4. Функция `println()` имеет один параметр. Она выводит переданное значение на экран и ничего не возвращает.

В основной ветке программы вызовите первую функцию. То, что она вернула, передайте во вторую функцию. Если вторая функция вернула `True`, то те же данные (из первой функции) передайте в третью функцию, а возвращенное третьей функцией значение – в четвертую.

По желанию. Если сделаете, загрузите на Ваш GitHub, на ТУИС РУДН прикрепить GitHub ссылку. Каждая работа будет оценена дополнительными баллами (1 балл за 1 задание), но срок выполнения на баллы такой же, как и у основных.

Задание 01

В программировании можно из одной функции вызывать другую. Для иллюстрации этой возможности напишите программу по следующему описанию.

Основная ветка программы, не считая заголовков функций, состоит из одной строки кода. Это вызов функции `test()`. В ней запрашивается на ввод целое число. Если оно положительное, то вызывается функция `positive()`, тело которой содержит команду вывода на экран слова "Положительное". Если число отрицательное, то вызывается функция `negative()`, ее тело содержит выражение вывода на экран слова "Отрицательное".

Задание 02

Напишите функцию

```
void stats(double* a, int size, double* min, double* max, double* avg);
```

которая вычисляет минимальное, максимальное и среднее значение элементов массива `a`. Результат она должна разместить соответственно по адресам `min`, `max`, `avg`.

Задание 03

Напишите функцию, которая удаляет из введенной пользователем строки все повторяющиеся символы и пробелы. После чего выводит полученный результат на экран. Например, если было введено "abc cde def", то должно быть выведено "abcdef".

Задание 04

Напишите программу, которая запрашивает ввод двух значений. Если хотя бы одно из них не является числом, то должна выполняться конкатенация, т. е. соединение, строк. В остальных случаях введенные числа суммируются.

Задание 05. Факториал-генератор

Напишите функцию-генератор возвращающую значение факториала. Эта функция должна работать по аналогии с функциями-генераторами языка Python: при каждом обращении к функции она возвращает следующее число Фибоначчи вплоть до некоторого его заданного номера. Напишите программу демонстрирующую работу этой функции.

Задание 06. «Ханойские башни»



Головоломка о ханойской башне была изобретена французским математиком Эдуардом Лукасом в 1883 году. Его вдохновила легенда, рассказывающая о замке Хинду, где эту задачу поставили перед юными жрецами. В начале времён им дали три стержня и стопку из шестидесяти четырёх золотых дисков, каждый из которых немного меньше предыдущего. Требовалось переставить все диски с одного стержня на другой, соблюдая два строгих условия. Во-первых, за раз можно было перемещать только один диск. Во-вторых, нельзя класть БОЛЬШЕЙ диск поверх меньшего. Жрецы работали (и работают по сей день) очень споро, день и ночь, переставляя каждую секунду по одному диску. Легенда гласит, что когда они закончат свою работу, замок обратится в пыль, и мир исчезнет.

Хотя легенда и интересна, вам не стоит беспокоиться о скором конце света. Число ходов, требующихся для правильной перестановки башни из 64 дисков, равняется $2^{64}-1=18,446,744,073,709,551,615$. Со скоростью один ход в секунду это займёт 584,942,417,355 лет! Большая цифра для такой несложной на первый взгляд головоломки.

Как нам решить эту задачу рекурсивно? С чего бы вы начали? Что является здесь базовым случаем? Давайте подумаем над этим от конца к началу. Предположим, изначально на первом колышке у вас находится башня из пяти дисков. Если вы уже знаете, как передвинуть четыре из них на второй колышек, то можете с лёгкостью переложить нижний диск на стержень №3, а затем переложить туда же башню со стержня №2. Но что, если вы понятия не имеете, как переместить башню из четырёх верхних? Предположим, что известно, как передвинуть башню из трёх верхних дисков на третий колышек. Тогда с перемещением четвёртого трудностей не возникнет: переложите его на второй стержень, а затем положите сверху те, что нанизаны на третий. Но если вы не знаете как переместить три? Что ж, можно переложить башню из двух дисков на стержень №2, а третий - на стержень №3, а потом сверху положить башню из двух. Но если до сих пор не понятно, как это сделать? Уверен, что вы согласитесь: переместить один диск на третий колышек легче лёгкого - тривиальнее ничего не найдётся. Звучит как базовый случай, а?

Вот общая схема того, как переместить башню с исходного стержня на заданный с использованием промежуточного:

Передвинуть башню из (количество дисков - 1) на промежуточный колышек, используя при этом заданный.

Положить оставшийся диск на заданный стержень.

Переместить башню из оставшихся на промежуточном стержне дисков на конечный, используя при этом первоначальный колышек.

Напишите функцию `move (n, x, y)`, которая печатает последовательность перекладываний дисков для перемещения пирамидки высоты `n` со стержня номер `x` на стержень номер `y`.

Реализуйте алгоритм в двух вариантах: на основе цикла и на основе рекурсии.