Лабораторная работа 2 по темам 5–7
(Раздел 3 – *Основы алгоритмизации*)

Цель выполнения лабораторной работы – закрепление теоретических знаний и получение навыков по темам раздела 3 «Основы алгоритмизации»:

Тема 5. Понятие и свойства алгоритма, способы записи алгоритмов
Тема 6. Формальное определение алгоритма
Тема 7. Вычислимые функции и методы разработки алгоритмов.

В ходе выполнения *проверяются следующие планируемые результаты обучения*:

- Студент *знает*: определение, свойства и способы формализации алгоритмов и методы исследования их характеристик, оценки эффективности; основные управляющие структуры и способы описания алгоритмов с использованием различных нотаций; основные методы разработки алгоритмов, особенности их реализации.
- Студент *умеет*: разрабатывать алгоритмы на основе простейшего базового набора функций, используя операции суперпозиции, примитивной рекурсии и минимизации; разрабатывать итерационные и рекурсивные алгоритмы, используя различные способы их описания; разрабатывать программы, реализующие рекурсивные и итерационные алгоритмы на языках высокого уровня (на языке C++).
- Студент *имеет навыки* чтения и понимания алгоритмов, описанных с использованием различных средств, пошагового выполнения алгоритмов («сухой прокрутки»).
- Студент *имеет представление* о способах оценки сложности алгоритмов.

При выполнении заданий разработанные алгоритмы должны быть описаны в виде блок-схем и/или структурограмм. Должны быть продемонстрированы навыки пошаговой детализации алгоритмов, использования различных управляющих конструкций при описании алгоритмов (ветвлений, циклов, ...).

Задания лабораторной работы 2

I. **Имеется множество вычислимых функций** (*базовый набор*):

$$P = \{ z(x), I_m^n(x_1, x_2, \ldots, x_n), Inc(x)\},$$

где $z(x) = 0$, $I_m^n(x_1, x_2, \ldots, x_n) = x_m$, $Inc(x) = S(x) = x + 1$.

Используя операции суперпозиции σ***, примитивной рекурсии*** ρ ***и минимизации*** μ***, постройте функции*** и покажите, как выполняются вычисления для заданных значений:

1. $Add(x, y) = x + y$ (покажите порядок рекурсивного вычисления для $x = 4$, $y = 2$).
2. $Mult(x, y) = x \times y$ (покажите порядок рекурсивного вычисления для $x = 4$, $y = 2$).
3. $Power(x, y) = x^y$ (покажите порядок рекурсивного вычисления для $x = 4$, $y = 2$).

Покажите ***порядок вывода формул и примеры вычислений построенных рекурсивных функций*** для заданных значений. При построении функций можно использовать полученные ранее функции, *расширяя ими базовый набор*. В примерах вычислений покажите *все* вызовы функций при прямом ходе рекурсии до функций приведённого выше *базового набора*.

Опишите рекурсивные и итерационные алгоритмы вычисления функций $Add(x, y)$, $Mult(x, y)$, $Power(x, y)$ ***в виде блок-схем и структурограмм***.

Разработайте на языке C++ функции, реализующие *a*) ***итерационные*** и *b*) ***рекурсивные*** алгоритмы вычисления приведённых выше функций ($Inc(x)$, $Add(x, y)$, $Mult(x, y)$, $Power(x, y)$). Выполните вычисления пошагово с помощью средств отладки, проверяя приведённые вычисления, – постройте трассировочные таблицы или деревья вызовов для выполнения каждой функции (используйте скриншоты) и сравните сложность рекурсивного и итерационного алгоритмов.

Примечание: Задачи 1, 2 и 3 частично разобраны на лекциях (см. презентацию по теме) и на практических занятиях (см. презентацию с примерами выполнения заданий) для других значений параметров. Используйте для выполнения заданий эти материалы.

II. ***Напишите*** *а*) ***рекурсивную функцию и*** *б*) ***функцию, реализующую итерационный алгоритм*** вычисления чисел Фибоначчи по заданному номеру, ***на языке*** **C++**:

$$Fib(n) = 1, \text{ если } n = 1 \text{ или } n = 2,$$
$$\text{иначе } Fib(n) = Fib(n-2) + Fib(n-1).$$

Выполните трассировку выполнения функций и/или <u>постройте дерево вызовов</u> для рекурсивного алгоритма со значением $n = 6$. Проверьте полученный результат (построенное дерево вызовов), выполнив программу в пошаговом режиме, используя средства отладки. <u>Используйте примеры из материалов лекции и практических занятий</u>.

III. ***Покажите порядок вычисления функции Маккарти при вызове функции со значением параметра*** $n = 98$ (<u>покажите вычисления, ***выполняя подстановки вызовов функции и результатов вычислений в строке***</u> по аналогии с примерами, разобранными на парах). Функция Маккарти («**Функция 91**») вычисляется по следующему правилу:

$$Маккарти(n) = n - 10, \text{ если } n > 100,$$
$$\text{иначе } Маккарти(n) = Маккарти(Маккарти(n+11)).$$

Напишите программу на языке C++, которая позволила бы ввести данные (число *n*) с клавиатуры и вычислить и вывести на экран значение функции Маккарти.
Проверьте полученный ранее результат (результат вычисления, выполненного вручную), выполнив программу в пошаговом режиме с заходом в функции. Проиллюстрируйте решение скриншотами пошагового выполнения (оформите трассировочную таблицу или постройте дерево вызовов).
Измените входные данные (возьмите значения больше 100 и меньше 100, например: $n = 101$, $n = 110$, $n = 90$ – выберите свои значения) и повторите вычисления вручную, проверьте результат, выполнив программу с вызовом функции для тех же значений.

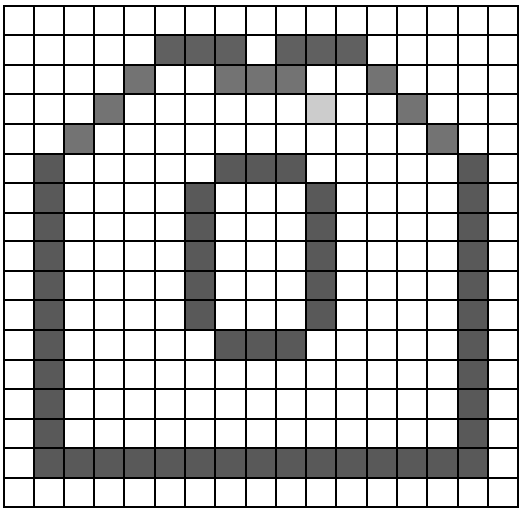
IV. ***Покажите порядок вычисления функции Аккермана при вызове функции со следующими значениями параметров:*** $n = 2$, $m = 1$ (покажите вычисления, <u>***выполняя подстановки вызовов функции и результатов вычислений в строке***</u> по аналогии с примерами, разобранными на парах). Функция Аккермана вычисляется по следующему правилу:

$$Аккерман(m, n) = n+1, \text{ если } m = 0 \text{ или}$$
$$\text{иначе } Аккерман(m, n) = Аккерман(m-1, 1), \text{ если } n = 0 \text{ или}$$
$$\text{иначе } Аккерман(m, n) = Аккерман(m-1, Аккерман(m, n-1)).$$

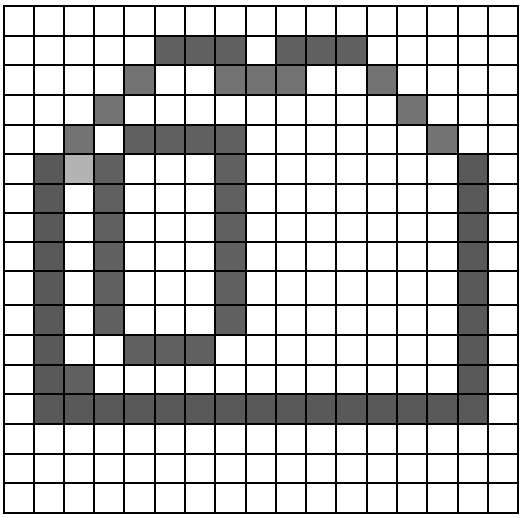
Напишите программу на языке C++, которая позволила бы ввести данные (числа *m* и *n*) с клавиатуры и вычислить и вывести на экран значение функции Аккермана. Проверьте полученный ранее результат (порядок вычисления вручную), выполнив программу в пошаговом режиме с заходом в функции. Проиллюстрируйте решение скриншотами пошагового выполнения (оформите трассировочную таблицу или постройте дерево вызовов).
Измените входные данные (выберите значения *m* и *n* самостоятельно) и повторите вычисления для выбранной комбинации значений.

V. ***Напишите рекурсивную процедуру закраски фигуры*** (текст процедуры на псевдокоде приведён в презентации с примерами для практических занятий). Используется два цвета (цвет границ фигур и цвет закраски). Покажите порядок закраски по шагам (постройте дерево вызовов и выполните «заливку» (*до 15 вызова процедуры минимум*), используя приведенную ниже схему). Начальная точка отмечена буквой "**x**" (первый вызов процедуры выполняется для точки с координатами, соответствующими выделенной на схеме клетке), каждой точке на экране соответствует элемент матрицы на схеме:

a) первый вариант:



b) второй вариант:



Примечание. При выполнении задания V можно не разрабатывать программу – для получения оценки 8 баллов за выполнение данного задания достаточно выполнить «сухую прокрутку» – построить дерево вызовов при указанных условиях и продемонстрировать по шагам, как выполняется заливка фигуры на каждом шаге. При разработке программы, если она реализуется на C++, использование графики можно заменить на «псевдографику»: фигуру для закраски представить матрицей, в ячейки которой вписываются символы (буквы например), обозначающие цвета границ и закраски. Процесс пошаговой закраски можно показать, выводя матрицу на экран после выполнения каждого шага алгоритма.

Примечание. При выполнении лабораторной работы используйте материалы лекций и практических занятий, а также проработайте вопросы и выполните задачи по темам раздела.

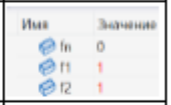
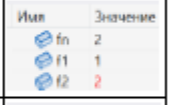
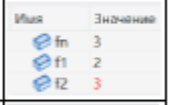
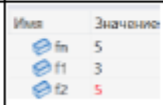
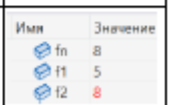
Примерные критерии оценки заданий

Отчёт о выполнении задания загружается в формате файла MS Word (.docx) или .ZIP, включает результаты выполнения всех заданий. Результаты выполнения каждого задания описываются под заголовком первого уровня. За титульным листом вставляется страница с оглавлением (содержанием) отчёта. Файл именуется с использованием фамилии и инициалов автора (например: ЛР-2-Иванов_ИИ.zip). Скриншоты в тексте должны быть сжаты для передачи по почте или публикации в Интернете.

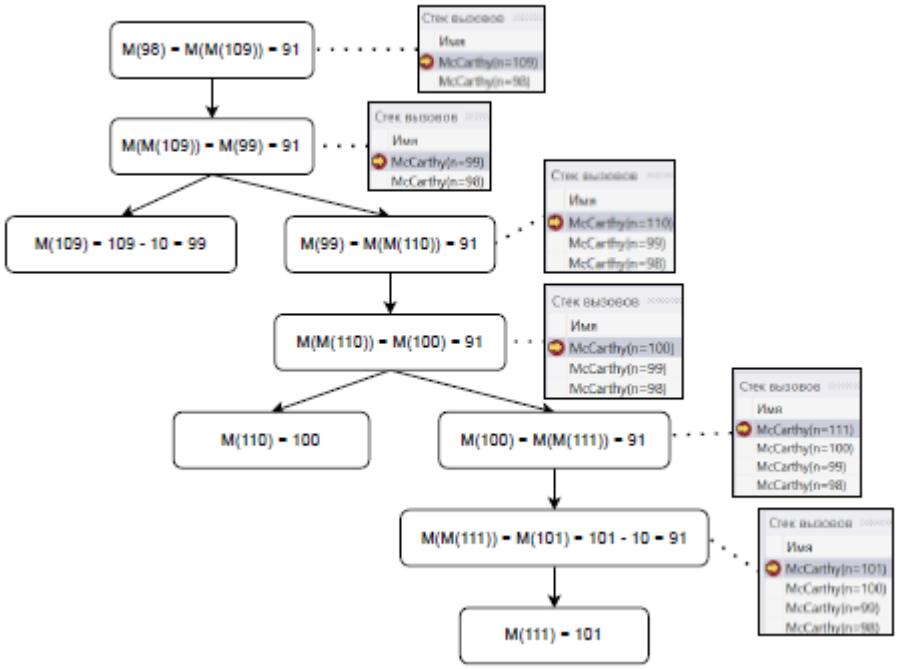
Все задания лабораторной работы (1–5) имеют одинаковый вес. Оценка за каждое задание определяется по следующим правилам:

Характеристика решения	**Оценка**
Задание выполнено полностью, с учётом всех требований. Приведены решения (там, где указано) для разных значений параметров функций, исходных данных. Все решения приведены в отчёте с соответствующими пояснениями по порядку вычислений. Представлены коды разработанных программ (функций) и приведено описание алгоритмов в виде блок-схем и структурограмм. Выполнено сравнение полученных результатов вычисления «вручную» и выполненных в программе. Выполнено сравнение сложности рекурсивных и итерационных алгоритмов (где требуется разработать оба варианта). В решениях нет ошибок	10
Задание выполнено полностью, с учётом всех требований. Имеются незначительные погрешности, неточности/пропуски в описании решений, формулировок, не приведено сравнение сложности алгоритмов	9
В отчёте отсутствует описание алгоритмов в виде блок-схем или структурограмм (приведён только один вариант описания алгоритмов). Все остальные требования выполнены	8
Результаты вычисления «вручную» проиллюстрированы построением дерева вызовов, трассировочными таблицами и т. п. (в зависимости от задания). Все предложенные в заданиях варианты вычислений выполнены. В отчёте не представлены результаты пошагового выполнения программ, сравнение результатов «ручного» вычисления функций и выполнения программ.	7
Результаты вычисления «вручную» проиллюстрированы построением дерева вызовов, трассировочными таблицами и т. п. (в зависимости от задания), но не все предложенные варианты вычислений выполнены. В отчёте не представлены результаты пошагового выполнения программ, сравнение результатов «ручного» вычисления функций и выполнения программ	6
Задания выполнены не полностью (например: приведены вычисления только для одного варианта входных данных, значений параметров; не разработаны итерационные или рекурсивные функции, где это требуется по заданию и т. п.), отсутствуют иллюстрации вычислений, выполненных «вручную» (не построено дерево вызовов, трассировочные таблицы и пр.), показаны только результаты выполнения разработанных программ	5
Выполнены основные требования для оценки 5 баллов, но в решениях имеются отдельные ошибки	4
Решения неполные, в решениях имеются ошибки – каждая ошибка снижает оценку на балл	1-3
Отчёт отсутствует	0

Пример оформления трассировочной таблицы с использованием скриншотов:

Операция	Значение *fn*	Комментарий	Количество итераций цикла	Скриншоты отладчика
FibNumber(n)		Вызов функции с аргументом $n = 6$		
$f1 = 1$ $f2 = 1$ $fn = 0$		Инициализация начальных значений		
		Начало работы цикла от 3 до n с шагом 1	0	
$fn = f1 + f2$ $f1 = f2$ $f2 = fn$	2	Вычисление fn как суммы двух предыдущих членов последовательности	1	
$fn = f1 + f2$ $f1 = f2$ $f2 = fn$	3	Вычисление fn как суммы двух предыдущих членов последовательности	2	
$fn = f1 + f2$ $f1 = f2$ $f2 = fn$	5	Вычисление fn как суммы двух предыдущих членов последовательности	3	
$fn = f1 + f2$ $f1 = f2$ $f2 = fn$	8	Вычисление fn как суммы двух предыдущих членов последовательности	4	
		Выход из цикла, возврат $fn = 8$		

Пример оформления дерева вызовов с использованием скриншотов:



Пример реализации «закраски» фигуры с использованием «псевдографики»:

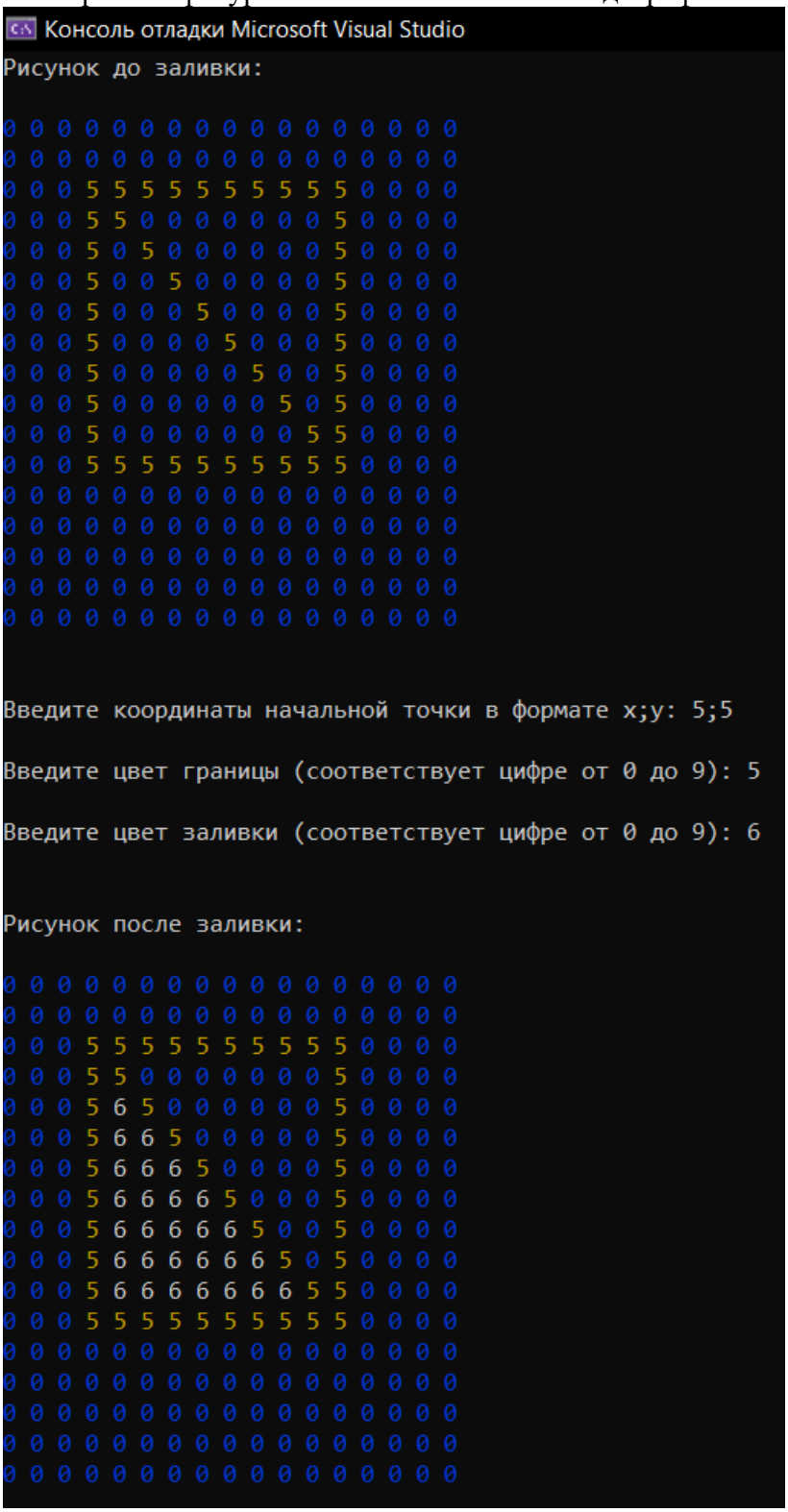


Иллюстрация дерева вызовов с использованием скриншотов: