Лабораторная работа 1. Рекурсия

Содержание

[Цель 2](#_Toc157539584)

[Порядок работы 2](#_Toc157539585)

[Требования в зависимости от оценки 8](#_Toc157539586)

# Цель

Необходимо реализовать программу для работы с рекурсиями и сделать отчет.

# Порядок работы

1. Напишите код программы и графический интерфейс, в которых можно анализировать работу рекурсии и циклических решений. Для своих вариантов задач, необходимо написать программы в которых решения будут выполняться с помощью циклов и с помощью рекурсии, замерять время выполнения обоих решений и выдавать пользователю. Также необходимо в интерфейсе иметь поле для ввода изначальных данных, а также меню с выводом формулы (для удобства пользователя). Также должна быть кнопка для запуска рекурсии и циклического решения с введенным значением в поле изначальных данных. И поле, которое отображает количество самовызовов функций в рекурсии (можно использовать глобальные переменные).

Варианты формул рекурсии:

* 1. F(n) = n, при n < 3;

F(n) = 2 \* (n - 1) + F(n - 1) + 2, если n > 2 и при этом n чётно;

F(n) = 2 \* (n + 1) + F(n - 2) - 5, если n > 2 и при этом n нечётно.

* 1. F(n) = n, при n < 3;

F(n) = 3 \* (n - 1) + F(n - 1) + 5, если n > 2 и при этом n чётно;

F(n) = 3 \* (n + 1) + F(n - 2) - 2, если n > 2 и при этом n нечётно

* 1. F(n) = 1, при n < 3;

F(n) = F(n - 1) + F(n - 2), если n > 2 и при этом n нечётно;

F(n) = ∑F(i) 1<=i<=n-1, если n > 2 и при этом n чётно,

* 1. F(n) = 1, при n < 3;

F(n) = F(n - 1) + F(n - 2), если n > 2 и при этом n нечётно;

F(n) = ∑F(i) 1<=i<=n-1, если n > 2 и при этом n чётно,

* 1. F(n) = 1, при n <= 1;

F(n) = 5 \* n + F(n - 1) + F(2), если n>1 и при этом n нечётно;

F(n) = 3 \* F(n - 1), если n > 1 и при этом n чётно.

* 1. F(n) = 1, при n <= 1;

F(n) = 4 \* n + F(n - 1) - F(2), если n>1 и при этом n нечётно;

F(n) = 3 \* F(n - 1), если n > 1 и при этом n чётно.

* 1. F(n) = 1, при n <=1;

F(n) = 3 + F(n - 1) \* F(n - 2) - F(n - 1) - F(n - 2), если n>1 и при этом n нечётно;

F(n) = 2 \* F(n - 1), если n > 1 и при этом n чётно.

* 1. F(n) = 2, при n <=1;

F(n) = 1 + F(n - 1) \* F(n - 2) - F(n - 1) - F(n - 2), если n>1 и при этом n нечётно;

F(n) = 2 \* F(n - 1), если n > 1 и при этом n чётно.

* 1. F(n) = 1, при n = 1;

F(n) = n + 2 \* F(n - 1) , если n чётно;

F(n) = 1 + 3 \* F(n - 2), если n > 1 и при этом n нечётно.

* 1. F(n) = 1, при n = 1;

F(n) = n + 3 \* F(n - 1) , если n чётно;

F(n) = 2 + 2 \* F(n - 2), если n > 1 и при этом n нечётно.

* 1. F(n) = 1, при n = 1;

F(n) = n + F(n - 1) , если n чётно;

F(n) = 2 \* F(n - 1) + F(n-2), если n > 1 и при этом n нечётно.

* 1. F(n) = 1, при n = 1;

F(n) = n + F(n - 1) , если n чётно;

F(n) = F(n - 1) + 2 \* F(n-2), если n > 1 и при этом n нечётно.

* 1. F(n) = 1, при n = 1;

F(n) = 1, при n = 2;

F(n) = 2 + F(n - 1), если n > 2 и при этом n чётно;

F(n) = 3 \* n + F(n - 2), если n > 2 и при этом n нечётно.

* 1. F(n) = 1, при n = 1;

F(n) = 1, при n = 2;

F(n) = 3 + F(n - 1), если n > 2 и при этом n чётно;

F(n) = 2 \* n + F(n - 2), если n > 2 и при этом n нечётно.

* 1. F(n) = 1, при n = 1;

F(n) = 2, при n = 2;

F(n) = [(7\*n + F(n - 3))/9], если n > 2 и при этом n четно

F(n) = [(5\*n + F(n - 1) + F(n - 2))/7], если n > 2 и при этом n нечетно

* 1. F(n) = 1, при n = 1;

F(n) = 2, при n = 2;

F(n) = [(8\*n + F(n - 3))/9], если n > 2 и при этом n четно

F(n) = [(4\*n + F(n - 1) + F(n - 2))/7], если n > 2 и при этом n нечетно

Каждый берет два варианта:

Первый: по вашему порядковому номеру в списке результатов, если вам не хватило, то определите вариант по формуле: Ваш\_порядковый\_номер % 16 +1.

Второй: Первый\_вариант + 8. Если получилось больше чем количество вариантов, то рассчитайте его по формуле: (Первый\_вариант + 8) % 16 +1.

1. Напишите еще одну программу с графическим интерфейсом (интерфейс может быть общим с предыдущей), которая будет выполнять задачу по вариантам (один вариант по номеру в таблице результатов), обратите внимание, что задания выполняются с помощью рекурсий:

|  |  |  |
| --- | --- | --- |
| № | Задание | Тестовые данные |
| 1 | Дано натуральное число N. Выведите слово YES, если число N является точной степенью двойки, или слово NO в противном случае.  Операцией возведения в степень пользоваться нельзя! |  |
| 2 | Дано натуральное число N. Вычислите сумму его цифр.  При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется). |  |
| 3 | Дано натуральное число N. Выведите все его цифры по одной, в обратном порядке, разделяя их пробелами или новыми строками.  При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется). Разрешена только рекурсия и целочисленная арифметика. |  |
| 4 | Дано натуральное число N. Выведите все его цифры по одной, в обычном порядке, разделяя их пробелами или новыми строками.  При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется). Разрешена только рекурсия и целочисленная арифметика. |  |
| 5 | Дано натуральное число n>1. Проверьте, является ли оно простым. Программа должна вывести слово YES, если число простое и NO, если число составное. Алгоритм должен иметь сложность . |  |
| 6 | Дано натуральное число n>1. Выведите все простые делители этого числа в порядке не убывания с учетом кратности. Алгоритм должен иметь сложность . |  |
| 7 | Дано слово, состоящее только из строчных латинских букв. Проверьте, является ли это слово палиндромом. Выведите YES или NO.  При решении этой задачи нельзя пользоваться циклами. |  |
| 8 | Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите все нечетные числа из этой последовательности, сохраняя их порядок.  В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция не возвращает значение, а сразу же выводит результат на экран. Основная программа должна состоять только из вызова этой функции. |  |
| 9 | Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите первое, третье, пятое и т.д. из введенных чисел. Завершающий ноль выводить не надо.  В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция не возвращает значение, а сразу же выводит результат на экран. Основная программа должна состоять только из вызова этой функции. |  |
| 10 | Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите наибольшее значение числа в этой последовательности.  В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция возвращает единственное значение: максимум считанной последовательности. Гарантируется, что последовательность содержит хотя бы одно число (кроме нуля). |  |
| 11 | Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите среднее значение элементов этой последовательности (без учета последнего нуля).  В этой задаче нельзя использовать глобальные переменные. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметра. В программе на языке C++ результат записывается в две переменные, которые передаются в функцию по ссылке.  Гарантируется, что последовательность содержит хотя бы одно число (кроме нуля). |  |
| 12 | Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите значение второго по величине элемента в этой последовательности, то есть элемента, который будет наибольшим, если из последовательности удалить наибольший элемент.  В этой задаче нельзя использовать глобальные переменные. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметра. В программе на языке C++ результат записывается в переменные, которые передаются в функцию по ссылке. Других параметров, кроме как используемых для возврата значения, функция не получает.  Гарантируется, что последовательность содержит хотя бы два числа (кроме нуля). |  |
| 13 | Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите, какое количество элементов этой последовательности, равны ее наибольшему элементу.  В этой задаче нельзя использовать глобальные переменные. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметра. В программе на языке C++ результат записывается в переменные, которые передаются в функцию по ссылке. Других параметров, кроме как используемых для возврата значения, функция не получает.  Гарантируется, что последовательность содержит хотя бы одно число (кроме нуля). |  |
| 14 | Дана последовательность натуральных чисел (одно число в строке), завершающаяся двумя числами 0 подряд. Определите, сколько раз в этой последовательности встречается число 1. Числа, идущие после двух нулей, необходимо игнорировать.  В этой задаче нельзя использовать глобальные переменные и параметры, передаваемые в функцию. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметров. |  |

Вариант по вашему порядковому номеру в списке результатов, если вам не хватило, то определите вариант по формуле: Ваш\_порядковый\_номер % 16 +1.

1. Нарисуйте блок–схему вашего приложения и добавьте ее в приложение А в отчете.
2. Напишите тестирование для вашего приложения и оформите его в приложении Б в отчете.
3. Оформите отчет по требованиям.
4. Сделайте архив с вашим проектом и отправьте его и отчет преподавателю.
5. Защитите работу на очном занятии.

# Требования в зависимости от оценки

1. Выполнены все пункты порядка работы;
2. Сделан отчет;
3. Отправлен проект;
4. В отчете продемонстрированы 5 блок-схем по требованиям ГОСТ;
5. Реализованы 3 unit-теста и 5 тест-кейсов;
6. Окно программы имеет верное название, соответствующее теме;
7. Корректное оформление отчета по всем требованиям;
8. Предусмотрена обработка ошибок.

Внимание: Оценка будет зависеть не только от требований к работе, но и от целостного качества всей работы.