

**Задание № 2 (часть 1)** Написание программы с использованием циклов (итерационные алгоритмы с переходным коэффициентом  $q$ )

Цель работы: освоение программирования циклов при реализации итерационных алгоритмов для закрепления принципов организации разного вида циклов

Пусть задана последовательность слагаемых  $u_0, u_1, \dots, u_n \dots$  с формированием последовательности частичных сумм  $S_0, S_1, \dots, S_n, \dots$ , где  $S_i = \sum_{j=0 \dots i} u_j$ .

Проводится вычисление некоторой частичной суммы  $S_n$ , где номер последнего слагаемого  $n$  определяется дополнительным требованием. Например, при заданном (обычно малом) числе  $\varepsilon > 0$  номер  $n$  может быть определен неявно соотношением  $n = \min\{i > 0 : |u_i| \leq \varepsilon\}$ .

Иначе говоря, для всех  $0 \leq i < n$ ,  $|u_i| > \varepsilon$ , а  $n$  — номер последнего слагаемого в частичной сумме  $S_n$  и  $|u_n| \leq \varepsilon$ . Очевидно, что соотношение  $|u_i| > \varepsilon$  можно использовать как условие продолжения цикла вычисления суммы.

В некоторых индивидуальных заданиях (при наличии неуклонного роста величины очередного слагаемого) это соотношение прямо противоположно — продолжения цикла вычисления суммы при  $|u_i| < \varepsilon$ .

На каждом шаге осуществляется страничный вывод на экран получаемых промежуточных значений (очередного слагаемого и частичной суммы) в **табличном формате** (с **вертикальными разделителями в строках**). Выход из цикла предусмотрен по достижении заданной точности или после выполнения тела цикла 1000 раз. **Используется потоковый ввод/вывод.**

### Основные этапы выполнения работы для практического задания № 2

1) понимание математической сущности задания на суммирование заранее неопределенного числа слагаемых — **переход от исходной сложной вычислительной формулы для очередного слагаемого к упрощенному выражению его вычисления на основе самостоятельно выведенной формулы с переходным коэффициентом:  $u_{i+1} = qu_i$**

2) **выполнение контрольного примера** — расчет по исходной формуле небольшой последовательности слагаемых (порядка 10-15) и **оценка особенности их поведения**, могут быть 2 варианта:

а) у большинства уменьшение на каждом шаге значения очередного слагаемого (его модуля) с остановкой при его достижении некоторого небольшого значения  $\varepsilon$  — задаваемого малого значения ( $0 < \varepsilon \ll 1$ ), определяющее точность нахождения решения;

б) несколько индивидуальных заданий имеют увеличивающейся (по модулю) вариант получения очередного слагаемого с остановкой при его достижении некоторого большого значения (для сходимости обозначим тем же  $\varepsilon$ ), превышающего некоторую величину у границы диапазона значений выбранного для переменных вещественного типа

3) **использование цикла do-while** при контроле вводимых данных

4) подготовка формата представления ВСЕХ рассчитываемых в программе значений с **выводом их в таблице с четко выделенными позициями** — организация вывода на экран

5) организация последовательного **вычисления слагаемых с использованием выведенной формулы** с помощью цикла while (или for)

### Требования к выполнению задания в программе № 1

1) выбрать в качестве данных для программы вещественные числа:

а) значение  $x$

б) значение  $\varepsilon$

в) значение очередных слагаемого и суммы

2) выбрать в качестве данных для программы целые числа: номер очередного слагаемого

- 3) определить условие окончания суммирования
- 4) явное вычисление степеней и факториалов не разрешается (за исключением ситуации возведения номера очередного слагаемого в степень с ее зависимостью от этого номера – например,  $i^i$ )
- 5) в дополнение к традиционному «приветствию» (ФИО, группа, даты) обеспечить **вывод на экран исходной расчетной формулы и выведенной расчетной формулы**
- 6) обеспечить проверку вводимых данных (минимум  $\varepsilon$ )
- 7) **сформировать таблицу с названиями столбцов** (содержит обозначение рассчитываемых программой величин) и строки с рассчитываемыми значениями, разделитель столбцов со значениями должен формировать строгую вертикальную линию – использовать манипуляторы для оформления таблицы
- 8) внимательно осмыслить выведенную расчетную формулу на предмет потенциального выхода за границы диапазона целых чисел **и при реализации программного кода видоизменить форму записи этой расчетной формулы** (советы приведены выше)
- 9) в отчет помещаются фрагменты из сформированного результата (начало и завершение таблицы, ее середина – в каждой части порядка 10 строк)

Оформляется «Отчет по практическому заданию № 2» с разделами:

1. Титульный лист (взять с сайта СПбГЭТУ для лабораторных работ)
2. Исходная формулировка задания
3. Математическая постановка задачи (разделы «Дано», «Найти», «Способ решения» с ПОДРОБНЫМ выводом расчетной формулы с использованием математической символики)
4. Контрольный пример (пошаговый расчет порядка 10-15 значений слагаемых ВРУЧНУЮ для конкретного НЕТРИВИАЛЬНОГО значения  $x$ ) с построением графика для выяснения варианта использования  $\varepsilon$
5. Организации интерфейса пользователя – разработка макетов ввода/вывода (четкое представление всех выводимых на экран сообщений с указанием предполагаемого формата оформления данных в виде таблицы), а также используемые возможности при вводе/выводе (команды/функции соответствующих библиотек)
6. Текст программы № 1
7. Результат работы программы (можно скриншотами)
8. Выводы

## Задание № 2 (часть 2) Написание программ с использованием массивов

Цель работы: продолжение освоения программирования циклов при реализации итерационных алгоритмов с использованием массивов для хранения рассчитываемых значений

### Особенности задания для лабораторной работы № 2

Используется выведенное по индивидуальному заданию в практическом задании № 2 соотношение для вычисления очередного слагаемого:  $u_i + 1 = q u_i$   
которое адаптируется под использование массивов:

- в программе № 2 для обычного массива через индексацию –  $u[i + 1] = q u[i]$
- в программе № 3 для динамического массива через указатели –  $*(p_{i+1}) = q *(p_i + i)$

Требуется адаптировать результат практического задания № 2 (написание программы № 1) по индивидуальному заданию для подготовки двух программ с использованием массивов по принципу «сначала рассчитываем значения с хранением в массивах, а потом отдельный вывод таблицы»:

- в программе № 2:
  - вводятся массивы с ограниченным числом элементов ( $N$ ) для хранения очередного слагаемого (массив  $A$ ) и очередной частичной суммы (массив  $S$ ),
  - массивы  $A$  и  $S$  последовательно заполняются сразу при каждом очередном шаге итерационного алгоритма,
  - ограничение работы итерационного алгоритма: по значению  $\varepsilon$  и по не превышению размера этих

массивов N,

- обращение к элементам массива осуществляется через индекс [i],
- вывод таблицы осуществляется ОТДЕЛЬНО после заполнения массивов по фактическому числу шагов,
- в итоге проход по массивам циклами осуществляется 2 раза (при заполнении при итерациях и при выводе таблице), итерационный алгоритм выполняется 1 раз;

– в программе № 3:

- предварительно определяется число шагов итерационного алгоритма (Step) по значению  $\epsilon$  и по некоторому предельному числу таких шагов (например, 1000) – формально соответствует циклу программы № 1 (но без вывода таблицы),
- выделяется с помощью указателей динамическая память размера Step+1 (не надо забывать про первое значение, формируемое до итераций) для хранения очередного слагаемого (динамический массив с доступом через указатель pA) и очередной частичной суммы (динамический массив с доступом через указатель pS),
- массивы через указатели pA и pS последовательно заполняются при каждом очередном шаге повторного выполнения итерационного алгоритма,
- ограничение второй работы итерационного алгоритма только по ранее вычисленного Step,
- обращение к элементам массива осуществляется через указатель  $*(p+i)$ ,
- вывод таблицы осуществляется ОТДЕЛЬНО после заполнения массивов согласно значения Step,
- в итоге проход по массивам циклами осуществляется 2 раза (при заполнении при итерациях и при выводе таблице), итерационный алгоритм выполняется 2 раз.

Естественно все запросы для ввода значений  $x$  и  $\epsilon$  и проверки вводимых значений сохраняются.

Результаты работы: сформированная таблица рассчитываемых значений (минимум – номер очередного слагаемого, его значение и накопленная частичная сумма), число слагаемых и зафиксированная сумма на последнем шаге.

### Требования к выполнению задания

1) выбрать в качестве данных для программы вещественные числа:

а) значение  $x$

б) значение  $\epsilon$

в) **массивы** для значений очередных слагаемого и суммы

2) выбрать в качестве данных для программы целые числа: номер очередного слагаемого

3) определить условие окончания суммирования

4) явное вычисление степеней и факториалов не разрешается (за исключением ситуации возведения номера очередного слагаемого в степень с ее зависимостью от этого номера – например,  $i^i$ )

5) в дополнение к традиционному «приветствию» (ФИО, группа, даты) обеспечить **вывод на экран исходной расчетной формулы и выведенной расчетной формулы**

6) обеспечить проверку вводимых данных (минимум  $\epsilon$ , а также  $x$  для вариантов с явным указанием на это)

7) **сформировать таблицу с названиями столбцов** (содержит обозначение рассчитываемых программой величин) и строки с рассчитываемыми значениями, разделитель столбцов со значениями должен формировать строгую вертикальную линию – использовать манипуляторы для оформления таблицы

8) внимательно осмыслить выведенную расчетную формулу на предмет потенциального выхода за границы диапазона целых чисел **и при реализации программного кода видоизменить форму записи этой расчетной формулы**

9) в отчет помещаются фрагменты из сформированного результата (начало и завершение таблицы, ее середина – в каждой части порядка 10 строк)

По программам № 2 и № 3 оформляется развернутый единый **«Отчет по лабораторной работе № 2»** с разделами:

1. Титульный лист (взять с сайта СПбГЭТУ для лабораторных работ)
2. Исходная формулировка задания
3. Математическая постановка задачи (разделы «Дано», «Найти», «Способ решения» с использованием математической символики) с расчетной формулой, взятой из **«Отчета по практическому заданию № 2»**
4. Особенности компьютерной реализации – раскрыть особенности использования двух вариантов работы с массивами: 1) объявленный ограниченный массив с определенным числом элементов с индексным обращением к ним ( $[i]$ ), 2) динамический массив с необходимым числом элементов и обращением к элементам через указатель (  $*(p+i)$  )
5. Организации интерфейса пользователя – разработка макетов ввода/вывода (четкое представление всех выводимых на экран сообщений с указанием предполагаемого формата оформления данных и формата ожидаемых для ввода значений), а также используемые возможности при вводе/выводе (команды/функции соответствующих библиотек) – существенно не отличается от описания в **«Отчете по практическому заданию № 2»**
6. Представление алгоритма решения задания в виде кратких словесных пояснений и ДВУХ блок-схем (для программы № 2 с ограниченным массивом и № 3 с динамическим массивом) с ПОЛНЫМ представлением ВСЕХ циклов (указать их условия и наполнение тел – можно номерами макетов и номерами расчетных формул, адаптированных под вид массива), а это do-while для проверки при вводе, while для итераций и for для вывода содержимого массивов в виде таблицы
7. Текст программ № 2 и № 3
8. Результаты работы программ № 2 и № 3 (можно скриншотами)
9. Выводы

Срок выполнения:

3 занятия по расписанию после выдачи задания (практические и лабораторное занятия 19, 20 и 21 января)