

Вариант 14

--- Задача 1 ---

Тема: Основные операции языка C.

Вычисление модуля и фазы комплексного числа. Написать программу, которая запрашивает у пользователя действительную и мнимую части комплексного числа, вычисляет и выводит на экран его модуль и фазу (в радианах).

***Примечание:** Все операции производятся в функции `main()`. Если в задании не указан явно тип операндов, предполагается использование вещественных чисел (тип `double`). В данном блоке заданий приведены условия, которые программа должна проверять при вводе пользовательских данных. Программа должна также выявлять и корректно реагировать на другие исключительные ситуации (например, деление на ноль). В последующих блоках заданий студентам предлагается определять эти условия самостоятельно, исходя из постановки задачи.*

--- Задача 2 ---

Тема: Основные операции языка C.

Определение максимального числа в последовательности положительных чисел. Пользователь вводит число элементов в последовательности, а затем её элементы. Последовательность состоит из положительных вещественных чисел. Программа определяет наибольший элемент в последовательности и выводит его значение и порядковый номер (порядковый номер соответствует первому из найденных чисел при наличии одинаковых наибольших элементов).

***Примечание:** Следует иметь в виду, что для данных с плавающей точкой операция равенства `"=="` не применяется, см. в WIKI "машинный эпсилон".*

--- Задача 3 ---

Тема: Функции.

Вычисление площади кольца. Написать функцию, которая вычисляет площадь кольца по значению внутреннего и внешнего радиусов. Радиусы передаются в качестве параметров. Считать ошибкой, если внешний радиус меньше внутреннего.

***Примечание:** Функция, которую требуется написать в каждом задании, не должна самостоятельно осуществлять ввод или вывод каких-либо значений, а также не должна прерывать выполнение программы. Ввод всех исходных данных, вызов функции, а также вывод результата и обработка ошибок осуществляются в функции `main()`. Вывод данных должен быть как можно более подробным и может включать не только возвращаемое функцией значение. В этом разделе и далее, следует учитывать, что внутри функции может произойти неисправимая ошибка (например, функции необходимо вычислить квадратный корень из отрицательного числа). В этом случае функция должна возвращать признак (код ошибки), для чего можно использовать дополнительный параметр с доступом по ссылке.*

--- Задача 4 ---

Тема: Одномерные массивы.

С одномерным массивом, состоящим из n вещественных элементов, произвести следующие операции:

- 1) найти количество элементов массива, равных 0;
- 2) вычислить сумму элементов массива, расположенных после минимального элемента.

При наличии нескольких минимальных элементов (равных друг другу) выбирается последний из них;

3) упорядочить элементы массива по возрастанию модулей элементов.

Примечание: Размеры массивов задаются именованными константами. Все операции над массивами выполняются отдельными функциями в том порядке, в котором они описаны в задании. Элементы массивов задаются одним из двух способов: генерация случайным образом или ввод с клавиатуры. Способ заполнения массива выбирается пользователем (необходимо предусмотреть соответствующий запрос). При составлении программ, максимальное количество элементов в массиве задается с помощью макроопределения `#define`. Однако программа должна спрашивать у пользователя реальную размерность массива (в диапазоне от единицы до константы в макроопределении). Для программ этого раздела, использовать глобальные массивы запрещается. Выход индекса за пределы массива считается грубейшей ошибкой.

--- Задача 5 ---

Тема: Двумерные массивы.

Дана целочисленная прямоугольная матрица. Упорядочить строки матрицы по возрастанию количества одинаковых элементов в каждой строке.

Примечание: Размеры массивов задаются именованными константами. Значения элементов массивов вводятся с клавиатуры, генерируются случайным образом или инициализируются в программе (значения по умолчанию). Способ задания элементов массива выбирается пользователем. Все операции над массивами выполняются отдельными функциями. Ввиду сложности передачи указателя на двумерный статический массив, рекомендуется создать глобальный массив и обращаться из каждой функции непосредственно к нему. Выход индекса за пределы массива считается грубейшей ошибкой. Особое внимание уделите порядку указания индексов в двумерном массиве.

--- Задача 6 ---

Тема: Динамически распределяемая память.

С одномерным динамическим массивом, состоящим из n вещественных элементов, произвести следующие операции:

- 1) вычислить сумму элементов массива с нечетными номерами;
- 2) вычислить сумму элементов массива, расположенных между первым и последним отрицательными элементами;
- 3) сжать массив, удалив из него все элементы, модуль которых не превышает 1. Размер блока памяти, выделенного под массив, изменить в соответствии с результатом сжатия.

Примечание: Размеры массивов и значения элементов вводятся пользователем. Все операции над массивами выполняются отдельными функциями в том порядке, в котором они описаны в задании. В каждой задаче необходимо предусмотреть функцию ввода с клавиатуры одномерного массива.

--- Задача 7 ---

Тема: Динамически распределяемая память.

Сложение двух матриц. Написать функцию, которая получала бы два двумерных динамических массива и возвращала бы указатель на двумерный динамический массив, каждый элемент которого равен сумме элементов двух исходных массивов.

Примечание: При работе со строками учитывать символ конца строки (байт с нулевым значением). Заполнение элементов массива (там, где это требуется) должно быть реализовано в виде отдельной функции. Во всех задачах, где необходимо вводить строки неограниченной длины без редактирования, следует иметь в виду, что для хранения строки необходимо использовать динамическую память. Размер блока, выделенного под строку, увеличивается при вводе очередного символа. Строка генерируется внутри функции ввода. Функция возвращает указатель на начало строки. Ввод строки завершается при нажатии клавиши Enter, ее код в строке не сохраняется.

Если на каком-либо этапе память выделить невозможно, то необходимо освободить всю выделенную ранее память и завершить программу. Особое внимание уделить обработке различных ошибок – при их возникновении необходимо освободить все ставшие ненужными ресурсы. При формировании нового массива можно реализовать любой из следующих алгоритмов, либо разработать свой: 1) создать новый массив и перенести в него нужные элементы. 2) скопировать исходный массив и поэлементно его изменять (например, переставлять элементы).

--- Задача 8 ---

Тема: Файлы.

Изменение данных в файле. В двоичном файле, содержащем данные типа unsigned int, возвести все элементы в квадрат. При превышении границы диапазона записать вместо элемента 0.

Примечание: Результат работы с данными вывести на экран и в новый файл. Файл с входными данными имеет название INPUT.TXT для текстовых файлов и INPUT.DAT для двоичных. Файл, в который записывается результат, называется OUTPUT.TXT для текстовых файлов или OUTPUT.DAT для двоичных. Число элементов определяется при считывании файла. Лучшими решениями считаются те, где нет привязки к количеству элементов в файле.

--- Задача 9 ---

Тема: Файлы.

Заполнение файла случайными числами и его последующая обработка. Заполнить текстовый файл целыми числами, полученными с помощью генератора случайных чисел. Найти количество четных чисел среди элементов файла с нечетными номерами.

--- Задача 10 ---

Тема: Решение дифференциального уравнения методом Эйлера.

Написать программу, находящую решение дифференциального уравнения $y' = f(x, y)$ методом Эйлера на отрезке $[x_0, x_n]$ с шагом h при начальных условиях $y = y_0$. Исходные данные приведены.

Метод Эйлера и вычисление функции $f(x, y)$ необходимо реализовать в виде отдельных функций. Исходные данные y_0 , x_0 , x_n , h вводятся с клавиатуры. Значения численного решения рассчитываются по формуле

$$y_{i+1} = y_i + h * f(x_i, y_i)$$

Исходные данные:

Дифференциальное уравнение: $y' = (\ln(x) - y) / x$

$$y_0 = -2$$

$$x_0 = 1$$

$$x_n = 2$$

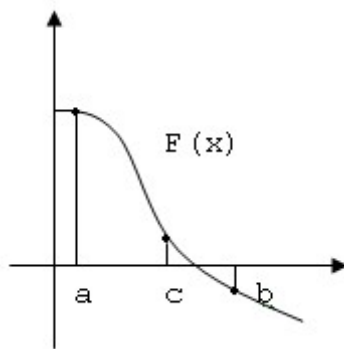
Точное решение ДУ: $y(x) = \ln(x) - 1 - 1/x$

--- Задача 11 ---

Тема: Решение трансцендентных уравнений

Написать программу, находящую решение уравнения $f(x)=0$ методом половинного деления на указанном отрезке изоляции. Точность вычисления корней уравнения $Eps=10^{-6}$. Исходные данные приведены. Метод половинного деления реализовать в виде функции. Поиск решения производить на отрезке $[-10;10]$. Выполнить приближенное вычисление с точностью Eps корня уравнения $f(x)=0$ при условии, что $f(x)$ непрерывна на $[a,b]$ и $f(a)*f(b)<0$, можно по следующей схеме:

1. Задать концы отрезка a и b , функцию f , малое число $Eps>0$ (допустимую абсолютную погрешность корня или половину длины его промежутка неопределенности); вычислить $f(a)$.
2. Вычислить $c = (a+b)/2$.
3. Если $b-a < 2Eps$, положить $x=c$ (x – корень уравнения) и остановиться.
4. Вычислить $f(c)$.
5. Если $f(c)=0$, положить $x=c$ и остановиться.
6. Если $f(a)*f(c)<0$, положить $b=c$ и вернуться к шагу 1; иначе положить $a=c$ и вернуться к шагу 1.



Алгебраическое уравнение: $2x^3 - x^2 + 1.9908x + 0.0184 = 0$

--- Задача 12 ---

Тема: Нахождение определенного интеграла

Написать программу, которая находит определенный интеграл $\int_a^b f(x)dx$ методом прямоугольников, трапеций и методом Симпсона. Исходные данные приведены. Шаг интегрирования взять равным $h=0,01$. Каждый метод должен быть реализован в виде отдельной функции.

Интеграл методом левых прямоугольников вычисляется по следующей формуле:

$$S = h \sum_{i=0}^{n-1} f(x_i)$$

Метод трапеций:

$$S = h \left(\frac{f(x_0) + f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i) \right)$$

Метод Симпсона:

$$S = \frac{h}{3} \sum_{i=0}^{\frac{n}{2}-1} \{ f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2}) \}$$

где $x_i = a + i*h$.

Исходные данные:

Функция $f(x) = (0.1 + x^{1/2}) / (1 + \ln^2 x)$

Отрезок интегрирования: [2;4]

--- **Задача 13** ---

Тема: Строки

Дана последовательность, содержащая до 5 слов, в каждом из которых до 5 строчных латинских букв; между соседними словами — не менее одного пробела, за последним словом — точка. Напечатать те слова последовательности, которые отличны от последнего слова и удовлетворяют следующему свойству:

- 1) буквы слова упорядочены по алфавиту;
- 2) последняя буква слова входит в него еще раз.

Примечание: Все операции над строками выполняются отдельными функциями в том порядке, в котором они описаны в задании.

Следует различать лексикографическое упорядочение — по коду символа, и упорядочение по алфавиту, в этом случае заглавные и строчные буквы располагаются рядом.