

БЛОК 1. Основы языка C.

Неделя 1. (Вводное занятие)

- Рассказ о плане занятий. Среды разработки Qt / MSVS. Настройка среды и запуск программы.
- Запуск программы «Hello World!» / Сохранить проект себе на сетевой диск / флэшку.
- Переменные: целочисленные и с плавающей точкой, арифметические операции, вывод значений переменных.
- Особенности деления и приведение типов.
- Ввод и вывод значений переменных и текста.
- Оператор ветвления IF.

Разбор решения задачи «Программа решения линейного уравнения».

- Этапы создания проекта/программы
- Кодирование, компиляция и запуск
- Разбор типовых ошибок (ошибки компиляции, деление на 0)
- Сохранение проекта

Неделя 2.

- Оператор ветвления, блок-схемы, условное исполнение. Форматирование кода.
- Алгоритм решения задачи – блок-схема
- Базовый оператор цикла FOR. Представление в виде блок-схемы.
- Операторы цикла WHILE и DO-WHILE как частные случаи FOR.
- Форматирование кода на примере вложенных циклов.
- Подключение математической библиотеки. Функция извлечения квадратного корня.

1. Программа решения квадратного уравнения (усвоение оператора ветвления и контроля типов).

- целочисленные коэффициенты задаются пользователем с клавиатуры
- разветвление решения на случай линейного и квадратного уравнения
- контроль корректности деления и извлечения квадратного корня
- отсутствие дублирующих математических операций (не надо дважды вычислять квадратный корень числа или значение корня если требуется только поменять знак)

2. Печать всех простых чисел не превышающих N (усвоение вложенных операторов цикла)

- число N задается пользователем с клавиатуры
- точная реализация математического определения: «имеет 2 делителя» (посчитать количество делителей)
- собственная реализация проверки делимости через приведение типов

Неделя 3.

- Отладочная печать как метод отладки программ.
- printf() – контроль количества выводимых знаков после запятой.

3. Вычислить число π с заданной точностью (кол-во знаков после запятой), используя ряд Грегори

- вычисления суммы бесконечного ряда, не зная количества необходимых членов (бесконечный цикл с выходом по критерию)
- собственная реализация правила округления чисел
- вывод количества потребовавшихся членов ряда для достижения точности
- вывод заданного числа знаком после запятой (параметры printf())

Неделя 4.

- Локальные и глобальные переменные. Пересечение имен.
- Объявление и реализация собственных функций. Параметры функции – локальные переменные.

4. Программа для решения уравнения вида $F(x) = 0$ методом Ньютона.

- Функция и ее производная задаются в тексте программы в виде отдельных функций а программе
float Func(float x) и float dFunc(float x)
- Начальное приближение и требуемая точность вычисления (количество знаков) задаются с клавиатуры

Неделя 5.

- Использование отладчика. Пошаговое исполнение как метод отладки программ.
- Контроль значений переменных во время отладки программ.

5. Вычисление интеграла функции $F(x)$ методом трапеций

- Функция задается в тексте программы в виде отдельной функции float Func(float x)
- шаг (количество разбиений) задается пользователем с консоли.
- Оцените точность вычисления

Неделя 6.

- Статические массивы данных.
- Директивы препроцессора. Глобальные константы и #define
- Указатели. Арифметика указателей.
- Выделение и освобождение памяти

6. Программа по вычислению максимума, минимума, среднего значения, среднеквадратичного отклонения во введенном статическом массиве

- максимальный размер массива задается #define, а актуальный размер – пользователем с клавиатуры при запуске
- хранение массива в виде глобальной переменной
- целочисленные элементы массива задаются пользователем с клавиатуры
- написание собственных функций Min(), Max(), Mean(), RMS()
- используйте массивы и оператор [] для доступа к элементам

БЛОК 2. Структуры данных, арифметика указателей, динамическая память, символы, строки.

Неделя 7-8.

- Многомерные массивы данных.
- Структуры данных. Использование Typedef
- Работа с файлами. Запись /чтение
- Методы определения конца файла

7. Работа с матрицами 3x3

- Структура Matrix_3x3 для хранения данных
- Данные хранятся в виде статического двумерного массива (float data[3][3])
- Функции вычисления суммы, произведения, определителя и обратной матрицы
- Исходная матрица заполняется целыми числами пользователем с экрана либо из файла.
- Результат операций выводится на экран и одновременно сохраняется в файл

Неделя 9.

- Указатели, динамическая память.
- Указатели на функции

8. Программа по вычислению максимума, минимума, среднего значения, среднеквадратичного отклонения во введенном динамическом массиве

- размер массива задается пользователем с клавиатуры при запуске
- целочисленные элементы массива задаются пользователем с клавиатуры
- выделение и освобождение памяти в функции main()
- используйте арифметику указателей и оператор * для доступа к элементам

Неделя 10.

- Алгоритм сортировки пузырьком.
- Оптимизации

9. Сортировка введенного динамического массива

- реализация алгоритма сортировки пузырьком
- направление сортировки задает пользователь с клавиатуры при запуске
- реализация единой функции сортировки в соответствии с математической моделью, за счет выделения функций Compare() и Replace()
- реализуйте в программе несколько функций сравнения для различных типов сортировки (по убыванию, по возрастанию, по абсолютному значению). Передайте указатель на соответствующую функцию сравнения в единую функцию сортировки

Неделя 11-13.

- Рекурсивные функции
- Генератор случайных чисел.

10. Работа с матрицами NxM

- Структура Matrix_NxM для хранения данных
- Данные хранятся в виде одномерного динамического массива (реализовать вычисление индекса элемента по его двумерному индексу в матрице).
- Функция InitMatrix() задает размер и выделяет память.
- Функция PrintMatrix() выводит матрицу на экран в табличном виде

- Функции вычисления суммы, произведения, определителя и обратной матрицы (функции не печатают результат, а возвращают его)
- Исходная матрица должна заполняться из файла или случайными числами
- Вычисление определителя с использованием рекурсивной функции (разложение по строке)

Неделя 14-16.

- Символьный тип `char`. Таблица ASCII. Ввод / вывод (`%c`)
- Строка как массив символов. `'\0'`. (спецификатор `%s`)

11. Обработка текста введенного пользователем

- Текст вводится пользователем при запуске программы
- текст хранится в статическом массиве заданного размера (ограничение на размер вводимого текста)
- реализация функции коррекции текста:
 - 1) большая буква только в начале предложения или в начале текста,
 - 2) удаление нескольких пробелов, следующих подряд,
 - 3) удаление пробелов перед точкой и запятой,
 - 4) добавление пробела после знаков препинания.
- проверка корректности размера массива (при добавлении пробелом не должен быть превышен размер массива)

12. Реализация функции ввода текста произвольного размера: `char* GetText(FILE* stream, char terminator)`

- Конец текста обозначается выбранным символом `terminator` (например, `'#'` или `'\n'`)
- Возвращается строка в стиле C (нуль-терминированная), не включающая символа `terminator`
- Выделение точного количества необходимой памяти (собственная функция перевыделения памяти без использования `realloc()`)
- Использование функции корректировки из задачи 9
- Ввод текста из консоли или файла (выбор задается с консоли)

БЛОК 3. Дополнительные главы.

- Списки данных. Односвязный список. Добавление элементов, удаление и перестановка.
- Заполнение заданным количеством препятствий в случайных местах с помощью генератора случайных чисел.
- Интерактивность: реагирования на действия пользователя.
- Дополнительные возможности языка и особенности структурного подхода в программировании
- Использование битовых операций
- Основные подходы для шифрования данных
- Основные подходы и алгоритмы для сжатия данных без потерь
- Текстовая графика

13. Записная книжка

- Список из структур данных `CPeople`
- Поля данных: имя, фамилия, номер телефона, дата рождения
- Функции добавления элемента, удаления (по имени-фамилии), сортировки (по любому полю)
- Собственная реализация функции сравнения строк
- Реализация «пользовательского меню» (действие выбирает пользователь с консоли)
- Функция сохранения списка в файл и загрузки из файла (с добавлением либо заменой по выбору пользователя)
- Основные черты интерфейса:

Критерии оценки:

«Удовлетворительно»
«Хорошо»
«Отлично»

задачи 1-7
требования на «3» + задачи 8-10
требования на «4» + задачи 11-13