

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

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

- N задается пользователем с клавиатуры
- точная реализация математического определения: «имеет 2 делителя»
- собственная реализация проверки делимости через приведение типов
- методы оптимизации вычисления: 0 делителей, ограничение цикла.

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
- собственная реализация правила округления
- проверка критерия остановки счета
- вывод количества потребовавшихся членов ряда

Неделя 4.

- Локальные и глобальные переменные. Пересечение имен.
- Объявление и реализация собственных функций. Параметры функции – локальные переменные.
- Использование отладчика. Пошаговое исполнение как метод отладки программ.
- Контроль значений переменных во время отладки программ.

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

- Функция F(x) задается преподавателем
- Функция и ее производная задаются в тексте программы в виде отдельных функций
float Func(float x) и float dFunc(float x)
- Начальное приближение и требуемая точность вычисления задаются с клавиатуры
- Проверьте свой ответ, используя открытые ресурсы в Интернете

5. Вычисление интеграла функции F(x) методом трапеций

- Используется та же функция, что и в задаче 4
- Функция задается в тексте программы в виде отдельной функции float Func(float x)

- шаг (количество разбиений) задается пользователем с консоли.
- Оцените точность вычисления
- Проверьте свой ответ, используя открытые ресурсы в Интернете

Неделя 5.

- 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 размер – пользователем
- хранение массива в виде глобальной переменной
- целочисленные элементы массива задаются пользователем с клавиатуры
- написание собственных функций, передача параметров, возвращение значений
- используйте указатели и оператор * в функциях для расчёта минимума и максимума (Min, Max)
- используйте массивы и оператор [] в функциях для расчёта среднего и среднекв. отклонения (Mean, RMS)

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

- размер массива задается пользователем
- целочисленные элементы массива задаются пользователем с клавиатуры
- выделение и освобождение памяти в функции main()
- перепишите Min, Max из предыдущей задачи через [], а Mean, RMS – через *

Неделя 6.

- Алгоритм сортировки пузырьком.
- Оптимизации

8. Сортировка введенного динамического массива

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

БЛОК 2. Символы, строки, структуры данных и списки.

Неделя 7.

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

9. 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
- Добавьте возможность чтения матрицы из файла (название задает пользователь с экрана; можно использовать функцию из задачи 10)
- Результат операций выводится на экран и сохраняется в файл

Неделя 8.

- Рекурсивные функции
- Генератор случайных чисел.

10. Работа с матрицами NxM

- Структура Matrix_NxM для хранения данных
- Данные хранятся в виде динамического массива.
- Функция InitMatrix задает размер и выделяет память.
- Функция PrintMatrix выводит матрицу на экран в табличном виде
- Функции вычисления суммы, произведения, определителя и обратной матрицы
- Исходная матрица должна заполняться из файла или случайными числами
- Результат операций выводится на экран и сохраняется в файл

- Вычисление определителя с использованием рекурсивной функции (разложение по строке)

Неделя 9.

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

11. Обработка текста введенного пользователем

- 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.

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 (задается с консоли)

БЛОК 3. Битовые операции, алгоритмика и проекты.

Неделя 10-11.

- Списки данных. Односвязный список. Добавление элементов, удаление и перестановка.

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:

Неделя 12-16

- Текстовая графика (библиотека CONLIB): позиционирование, перерисовка, управление цветом.
- 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 (**доклад студентов**)
- Основные подходы для шифрования данных (**доклад студентов**)
- Основные подходы и алгоритмы для сжатия данных без потерь (**доклад студентов**)
- Текстовая графика (**доклад студентов**)

14. Проект по согласованию с преподавателем

Примеры проектов:

- Бродилка в лабиринте
- Парсинг математических выражений и символьное дифференцирование
- Архиватор (Huffman coding, ...)
- Игра (Тетрис, Сапёр, ...)
- Библиотека для работы с длинной арифметикой
- Реализация нетривиальных алгоритмов (ассоциативные массивы на деревьях, работа с графами, ...)

Критерии оценки:

«Удовлетворительно»	задачи 1-9
«Хорошо»	требования на «3» + задачи 10-12
«Отлично»	требования на «4» + 13 + 14

- В конце семестре предлагается для желающих 3 доклада на темы 12-16 недель. Допускается работа в парах. Успешный доклад засчитывается как +0,5 задачи (к соответствующей оценке).