

# Язык программирования Си

Бикулов Д.А., Иваницкая Н.В., Иванов А.П.

## Семинар 9. Зачет: методика его проведения, теоретический минимум и типовые задания.

Зачет проводится в практикуме и состоит из двух частей: теоретической и практической.

### 1 Теоретический минимум

Проверка теоретических знаний производится в виде опроса по любым вопросам, входящим в программу первого семестра и теоретический минимум. Вопрос, как правило, формулируется в виде просьбы написать на листе бумаги короткий (2-5 строк) код, иллюстрирующий то или иное понятие языка программирования. Обычно задается от двух до пяти вопросов. Студенты, не сдавшие теоретическую часть, автоматически не допускаются к выполнению практической части зачета.

Ниже приводится перечень сведений о языке программирования из теоретического минимума знаний, которые студент должен иметь для получения зачета.

1. Основы синтаксиса языка Си, структура консольного приложения.
2. Фундаментальные типы данных (`bool`, `char`, `int`, `double`).
3. Определение переменных и констант.
4. Оператор `sizeof()`.
5. Выражения, операции, комментарии.
6. Оператор приведения типа.
7. Операторы инкремента и декремента.
8. Приоритет операторов в выражениях.
9. Блоки и правила видимости переменных.
10. Условный оператор и оператор перехода (`if`, `goto`).
11. Оператор множественной альтернативы (`switch`).
12. Цикл `while`. Прерывание цикла. Переход к следующей итерации.
13. Цикл `do ... while`. Прерывание цикла. Переход к следующей итерации.
14. Цикл `for`. Прерывание цикла. Переход к следующей итерации.
15. Математические функции стандартной библиотеки Си (`<math.h>`).
16. Форматированный консольный ввод (`<stdio.h>`): параметры функции `scanf()`.
17. Форматированный консольный вывод (`<stdio.h>`): параметры функции `printf()`.
18. Форматированный файловый ввод-вывод (`<stdio.h>`).
19. Бесформатный файловый ввод-вывод (`<stdio.h>`).
20. Массивы. Передача массивов в параметрах функции.
21. Определение функции. Прототип функции. Рекурсия.
22. Параметры функции `main()`.

23. Раздельная компиляция программных модулей. Использование \*.h файлов.
24. Внешние (**extern**) и глобальные переменные.
25. Статические (**static**) переменные.
26. Статические (**static**) функции.
27. Указатели и операторы, с ними связанные.
28. Указатель на функцию.
29. Функции для работы с динамической памятью **malloc()**/**realloc()**/**free()**.
30. Строки Си. Функции для работы со строками (**<string.h>**).
31. Функции для работы с символами (**<ctype.h>**).
32. Пользовательский тип данных **enum**.
33. Пользовательский тип данных **struct**.
34. Пользовательский тип данных **union**.
35. Определение синонимов типов (**typedef**).
36. Директивы препроцессора для условной компиляции и их использование.
37. Директивы препроцессора для включения файлов и их использование.
38. Макроопределения препроцессора (с параметрами и без).

## **2 Вопросы по теоретической части**

Здесь приводятся примерные варианты билетов с вопросами к теоретическому минимуму, на которые студент должен уметь отвечать на зачете.

---

### **1. Вариант**

1. Какие базовые типы данных Вы знаете? Сколько места они занимают в памяти компьютера?
  2. Что означает команда «**continue**» и где она применяется?
  3. Выделите память под вещественный массив чисел двойной точности из 123 элементов.
- 

### **2. Вариант**

1. Какие операторы языка Си Вы знаете? Какой у них приоритет?
  2. Какие виды циклов Вы знаете? Чем они отличаются друг от друга?
  3. Что такое прототип функции и где он используется?
- 

### **3. Вариант**

1. Что такое операции постфиксного инкремента и префиксного декремента?
2. Что означает команда «**break**» и где она применяется?
3. Что такое рекурсивная функция? Приведите пример.

#### 4. Вариант

1. Что означает строка:

```
void func(int, int*);
```

2. Что такое метка и где она используется?
  3. Что такое цикл «с предусловием» и цикл «с постусловием»?
- 

#### 5. Вариант

1. Сколько места в памяти компьютера занимают вещественные переменные и переменные с двойной точностью? Что такое точность представления вещественного числа?
  2. Что означает команда «goto» и где она применяется?
  3. Как найти заданную подстроку в другой строке?
- 

#### 6. Вариант

1. Что такое массив и что такое указатель? Приведите примеры.
2. Верно ли синтаксически написана программа:

```
void main( ) { }
```

3. Какие функции из библиотеки <string.h> Вы знаете?
- 

#### 7. Вариант

1. Что такое операция получения адреса? Где она используется?
  2. Какие функции ввода-вывода Вы знаете?
  3. Что такое перечислимый тип? Как он определяется?
- 

#### 8. Вариант

1. Как получить значение по указателю? Приведите примеры.
  2. Какие форматирующие последовательность функции `printf()` Вы знаете?
  3. Опишите структуру, состоящую из символьного массива из 12 членов, целого и вещественного числа.
- 

#### 9. Вариант

1. В каком случае можно изменить размер ранее выделенного массива? Как это сделать?
  2. Напишите цикл `do ... while()`, меняющий порядок элементов массива на противоположный.
  3. Для чего нужен оператор `typedef`?
- 

#### 10. Вариант

1. Напишите цикл `for`, меняющий порядок элементов массива на противоположный.
  2. Как получить указатель на переменную?
  3. Какие функции для работы с отдельными символами Вы знаете?
-

### 11. Вариант

1. Как записать массив в бинарный (неформатированный) файл?
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 1, b = 2, c = 3;  
c -= a += b = c;
```

3. Как вернуть из функции массив?
- 

### 12. Вариант

1. Откройте файл "myfile.txt" на чтение.
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 1, b = 2;  
a += b++;
```

3. Что такое `union`?
- 

### 13. Вариант

1. Как передать функцию в другую функцию? Как создать массив функций?
  2. Как с помощью функции языка Си прочитать в три переменные три вещественных числа, разделенных пробелами?
  3. Как обратиться к функции, определенной в другом файле с исходным текстом?
- 

### 14. Вариант

1. Что такое заголовочный файл?
  2. Как определить размер вещественного массива в байтах?
  3. Напишите функцию, которая складывает два вещественных числа и возвращает результат.
- 

### 15. Вариант

1. Для чего нужны статические переменные? Чем они отличаются от глобальных?
  2. Что такое макроопределение с параметром? Приведите пример.
  3. Как распечатать на экран строки из текстового файла, открытого на чтение?
- 

### 16. Вариант

1. Как найти длину текстовой строки?
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 1, b = 4, c = 3;  
a = --b > c;
```

3. Для чего применяется оператор `default`?
- 

### 17. Вариант

1. Как реализовать компиляцию некоторой функции в зависимости от условия?
  2. Напишите циклы `while`, транспонирующие матрицу.
  3. Опишите структуру, состоящую из имени, фамилии и года рождения студента.
-

### 18. Вариант

1. Определите двумерный вещественный массив.
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 1; int* b = &a;
(*b)++;
```

3. Определите массив из трех структур, которые представляют разные точки на плоскости.
- 

### 19. Вариант

1. Как изменить значение глобальной переменной, заданной в другом файле исходного текста проекта?
  2. Какие форматирующие последовательности функции `printf()` Вы знаете?
  3. Как задать перечислимый тип из трех убывающих элементов?
- 

### 20. Вариант

1. Выделить динамическую память под строку из 20 символов.
  2. Как передать аргументы с командной строки в функцию `main()`?
  3. Распечатайте число  $\pi$  с тремя знаками после десятичной точки.
- 

### 21. Вариант

1. Как закрыть функцию от использования в других исходных текстах проекта?
2. Какое значение будет иметь переменная `x`:

```
double x = 4;
x *= 1/2;
```

3. Определите функцию, которая возвращает сумму элементов переданного массива.
- 

### 22. Вариант

1. Как найти в строке любой из символов, входящих в строку "abc"?
  2. Как напечатать целое число с обязательным выводом знака «плюс» для положительных чисел? Как сделать выравнивание по левой границе поля для этого числа?
  3. Как принудительно преобразовать тип целого числа к вещественному типу?
- 

### 23. Вариант

1. Как получить адрес третьего элемента вещественного массива? Если этот адрес уменьшить на единицу – куда он будет указывать?
2. Какое значение получит переменная `z` в результате выполнения программы:

```
int a = 1, b = 2, c = 3, d = 4; bool z;
z = a < b || b < c && d < c;
```

3. Как определить, является ли заданный символ пробельным?
-

#### 24. Вариант

1. Создайте макроопределение, которое вычисляет сумму двух своих аргументов.
2. Что делает функция `scanf()`?
3. Откройте файл "`myfile.txt`" на запись в режиме «запись в конец файла».

---

#### 25. Вариант

1. Как проверить, открылся ли файл?
2. Какие значения будут иметь все переменные в результате выполнения программы:

```
int a = 0, b = 1, c = 1;  
a = b << c++;
```

3. Как обратиться к переменной, определенной в ином исходном тексте проекта?

### 3 Практические задания

Практические навыки программирования проверяются на одной типовой задаче, подобной тем, что выполнялись в течение семестра. Студент должен ее выполнить в течение одной пары (два академических часа) от начала до конца.

Рекомендуется основную массу заданий выдавать по образцу седьмого семинара: написать программу, которая читает данные (например, матрицу чисел произвольной размерности) из одного файла, как-то преобразует прочитанные данные (например, транспонирует прочитанную матрицу) и записывает получившийся результат в другой файл.

В приводимых ниже типовых заданиях для зачета необходимо написать законченную программу, которая тестирует заданную функцию.

---

#### 1. Вариант

Функция записывает в файл все четырехзначные натуральные числа из диапазона (2000 – 3000), в записи которых нет двух одинаковых цифр, подсчитывает количество таких чисел, возвращает в `main()`.

---

#### 2. Вариант

Функция считывает из файла вещественный массив неизвестной длины (до конца файла). Надо найти в массиве два элемента, модуль разности которых имеет наименьшее значение. Напечатать эти элементы и их индексы.

---

#### 3. Вариант

Написать функцию, работающую как простейший калькулятор, выполняющий действия «+», «-», «\*», «/» над двумя целыми массивами одного размера, считанными из файла. Результат записать в файл.

---

#### 4. Вариант

Функция считывает из файла целый массив неизвестной длины (до конца файла), находит в массиве минимальный по модулю элемент и заменяет им все элементы массива с четными номерами. Измененный массив записывается в файл.

---

#### 5. Вариант

Функция считывает из файла целый массив неизвестной длины (до конца файла). Необходимо найти максимальный по модулю элемент этого массива и заменить им все нулевые элементы массива. Подсчитать и вернуть в `main()` число таких элементов. Измененный массив сохранить в файл.

---

#### 6. Вариант

Функция считывает из файла целый массив неизв. длины (до конца файла). Определить и напечатать три наибольших элемента этого массива. Подсчитать их сумму, вернуть в `main()`.

---

#### 7. Вариант

Функция считывает из файла вещественный массив неизв. длины (до конца файла), находит в массиве минимальный по модулю элемент и заменяет им все элементы с нечетными номерами. Измененный массив записывается в файл.

---

### 8. Вариант

Функция получает из `main()` два целых числа,  $m$  и  $n$ , создает массив из простых чисел, расположенных в интервале от  $m$  до  $n$ , подсчитывает их количество, массив записывает в файл.

---

### 9. Вариант

Функция считывает из файла целый массив неизв. длины (до конца файла) и запрашивает у пользователя целое число  $k$ . Найти элемент массива, значение которого наиболее близко к введенному целому числу, его индекс вернуть в `main()`.

---

### 10. Вариант

Написать и протестировать функцию, вычисляющую методом трапеций площадь под графиком непрерывной положительной  $f(x)$  на отрезке  $[a, b]$ . Шаг разбиения уменьшать, пока площадь от итерации к итерации изменяется более, чем на  $\epsilon$ . Параметры  $a, b$  и  $\epsilon$  передаются из `main()`, куда и возвращается результат.

---

### 11. Вариант

Написать и протестировать функцию, вычисляющую методом верхних прямоугольников площадь под графиком непрерывной положительной  $f(x)$  на отрезке  $[a, b]$ . Шаг разбиения уменьшать, пока площадь от итерации к итерации изменяется более, чем на  $\epsilon$ . Параметры  $a, b$  и  $\epsilon$  передаются из `main()`, куда и возвращается результат.

---

### 12. Вариант

Написать и протестировать функцию, вычисляющую методом деления отрезка пополам с точностью  $\epsilon$  корень уравнения  $f(x) = 0$  на отрезке  $[a, b]$ . Параметры  $a, b$  и  $\epsilon$  передаются из `main()`, куда и возвращается результат. Функция  $f(x)$  вычисляется в отдельном блоке.

---

### 13. Вариант

Создать функцию, которая должна считать массив чисел неизвестной размерности из файла, отсортировать и записать в другой файл. Функция принимает название входного и выходного файла. Необходимо использовать функции выделения динамической памяти.

---

### 14. Вариант

Создать функцию, которая считывает из файла численные данные, записанные в 2 колонки, и переписывает их в другой файл в строку через запятую. Необходимо использовать функции выделения динамической памяти.

---

### 15. Вариант

Написать и протестировать функцию, которая находит в текстовом файле заданную пользователем строку и указывает номер строки файла, в которой строка была найдена.

---

### 16. Вариант

Создать функцию, подсчитывающую количество букв в тексте, находящемся в указанном файле. Функция принимает название входного файла и возвращает количество букв. Цифры не учитывать.

---

### 17. Вариант

Создать функцию, обрабатывающую текстовый файл. В файле имеется массив  $x$ -координат, размер которого заранее не известен. Функция считывает эти значения и записывает в выходной файл  $x$  и  $\sin(x/2)$  в 2 колонки. Необходимо использовать функции выделения динамической памяти.

---

### 18. Вариант

Создать функцию, обрабатывающую текстовый файл. В файле имеются две колонки вещественных чисел (пары координат  $x, y$ ). Надо отсортировать пары чисел по координате  $x$  и записать отсортированные таким образом две колонки в другой файл.

---

### 19. Вариант

Имеется 2 текстовых файла, в которых записаны два  $n$ -мерных вектора. Значение  $n$  заранее неизвестно. Создать функцию, которая считывает вектора из файлов, производит сложение или поэлементное векторов и записывает результат в 3-й файл. Функция принимает название входного и выходного файлов и знак операции («+» или «\*»). Необходимо использовать функции выделения динамической памяти.

---

### 20. Вариант

Написать функцию, перемножающую две матрицы вещественных чисел произвольного размера. Каждая матрица записана в своем текстовом файле. Функция должна считывать эти матрицы, перемножать их и записывать результат в третий файл. Функция принимает название входных и выходного файлов.

---

### 21. Вариант

Создать структуру, представляющую собой рациональное число (пара целых чисел: числитель и знаменатель). Написать функции для выполнения четырех арифметических действий с такими структурами. Написать тестовую программу, в которой нужно арифметическое выражение вводит пользователь.

---

### 22. Вариант

Написать программу, которая в текстовом файле при помощи пробелов и звездочек рисует окружность, заданного пользователем радиуса.

---

### 23. Вариант

Написать функцию, которая считывает из указанного ей файла матрицу произвольной размерности, считает среднее арифметическое и стандартное отклонение для каждой колонки этой матрицы и результат распечатывает на экране.

---

### 24. Вариант

Написать функцию, которая сортирует в алфавитном порядке переданный ей массив англоязычных текстовых строк. Строки считываются из указанного файла, а результат выводится на экран.

---

### 25. Вариант

Создать структуру, которая будет содержать вещественный вектор произвольной размерности. Реализовать функцию вставки нового элемента в любое место этого вектора.

---