Основные принципы и понятия языка С. Консольное приложение. Основные встроенные типы данных. Переменные, константы. Представление вещественных переменных в памяти

1. Какие из следующих типов данных являются встроенными в C?

a\*) int

b) string

c\*) double

d) class

2. Что такое переменная в программировании?

a\*) Именованное место в памяти, хранящее значение

b) Условие выполнения операции

c) Код, выполняющий определенную задачу

d) Программа, которая выполняет ввод и вывод данных

3. Как объявить константу в C?

a) Используя ключевое слово "var"

b) Используя знак равенства (=)

c) Используя ключевое слово "const"

d\*) Используя ключевое слово "const" и указав тип данных

4. Какой тип данных используется для хранения целых чисел в C?

a) float

b\*) int

c) char

d) bool

5. Какие из следующих операторов используются для ввода и вывода данных в консоли в C?

a) input() и output()

b) read() и write()

c\*) scanf() и printf()

d) cin и cout

6. Какие из следующих выражений являются правильным способом объявления целочисленной переменной?

a) integer x;

b\*) int x;

c) x = int;

d) int = x;

7. Какой тип данных используется для хранения вещественных чисел в C?

a) int

b) char

c) string

d\*) double

8. Какова размерность (в байтах) типа данных double в C?

a) 2

b) 4

c\*) 8

d) 16

9. Какие из следующих операторов используются для выполнения математических операций в C?

a) and и or

b) if и else

c\*) + и -

d) true и false

10. Как объявить и инициализировать переменную x в C, чтобы она содержала значение 10?

a) x = 10;

b\*) int x = 10;

c) int x;

d) x(10);

11. Что такое тип данных bool в C?

a) Целочисленный тип данных

b) Тип данных для хранения символов

c\*) Логический тип данных

d) Тип данных для хранения дробных чисел

12. Какое ключевое слово используется для определения условий в операторе if в C?

a) case

b\*) if

c) then

d) switch

13. Что означает выражение "sizeof(int)" в C?

a) Размер типа данных int в байтах

b) Количество бит в типе данных int

c\*) Размер типа данных int в байтах

d) Количество элементов типа данных int

14. Какие из следующих чисел являются вещественными числами?

a) 42

b) B5

c) -7

d\*) 0.5

15. Как объявить константу с плавающей запятой с точностью до 2 знаков после запятой в C?

a) const float pi(3.14);

b) float const pi = 3.14;

c) float pi = const 3.14;

d\*) const float pi = 3.14;

Основные операции языка С. Консольный ввод и вывод. Массивы.Условные операторы, циклы.

1. Как объявить массив целых чисел с 5 элементами в C?

a) int arr(5);

b\*) int arr[5];

c) array<int> arr(5);

d) int[5] arr;

2. Как получить значение элемента массива по его индексу в C?

a) arr(index)

b\*) arr[index]

c) arr.element(index)

d) arr.get(index)

3. Какой оператор используется для выполнения условных операций в C?

a) for

b\*) if

c) while

d) switch

4. Какой цикл используется для выполнения повторяющихся операций, пока условие истинно в C?

a) do-while

b) for

c\*) while

d) repeat-until

5. Какой оператор используется для увеличения значения переменной на 1 в C?

a) increment

b) add

c) plus

d\*) ++

6. Какой оператор используется для уменьшения значения переменной на 1 в C?

a) decrement

b) minus

c) reduce

d\*) --

7. Какая команда используется для выхода из цикла в C?

a) out

b) exit

c\*) break

d) continue

8. Какой цикл используется для выполнения повторяющихся операций с предусловием в C?

a) do-while

b) for

c) while

d\*) while

9. Какой оператор используется для выполнения операции целочисленного деления в C?

a) \*

b) div

c) %

d\*) /

10. Какой оператор используется для проверки двух условий, где оба условия должны быть истинными для выполнения действия?

a) or

b) not

c) nor

d\*) &&

11. Какой оператор используется для проверки двух условий, где хотя бы одно из условий должно быть истинным для выполнения действия?

a) and

b\*) ||

c) not

d) nor

12. Какой цикл используется, когда количество итераций заранее известно?

a) do-while

b\*) for

c) while

d) repeat-until

13. Какой оператор используется для выполнения определенной операции над каждым элементом массива в C?

a) apply

b\*) for

c) each

d) iterate

14. Какой оператор используется для выполнения операции целочисленного деления с остатком в C?

a) /

b) div

c\*) %

d) //

15. Какой оператор используется для выполнения кода многократно на основе условия в C?

a) for

b) if

c\*) while

d) switch

Функции в языке С. Перегруженные функции. Область действия переменных и связанные с ней понятия. Создание программ из нескольких модулей. Условная компиляция программ

1. Что такое функция в языке C?

a) Описание структуры данных

b) Часть оперативной памяти

c\*) Фрагмент кода, выполняющий определенную задачу

d) Графический интерфейс программы

2. Как объявить функцию в C?

a) function myFunction();

b) void myFunction;

c\*) int myFunction();

d) define myFunction();

3. Что такое перегруженные функции в C?

a) Функции, которые выполняют множество разных задач

b) Функции, которые имеют множество параметров

c) Функции, которые могут быть вызваны из разных модулей

d\*) Функции с одинаковыми именами, но разными параметрами

4. Что такое область действия переменных в C?

a) Место, где переменная объявлена

b) Время жизни переменной

c\*) Часть программы, где переменная может быть использована

d) Размер переменной в байтах

5. Как объявить локальную переменную внутри функции в C?

a) local x;

b) variable x;

c\*) int x;

d) x = 5;

6. Что такое глобальная переменная в C?

a) Переменная, которая видна только внутри функции

b\*) Переменная, которая видна во всей программе

c) Переменная, которая видна только внутри цикла

d) Переменная, которая видна только внутри условного оператора

7. Каким образом можно передать значение в функцию в C?

a) Через глобальные переменные

b) Только через параметры функции

c\*) Через параметры функции и глобальные переменные

d) Только через возвращаемое значение функции

8. Какие из следующих операторов используются для определения условий в C?

a) for и while

b\*) if и else

c) switch и case

d) do и until

9. Что такое условная компиляция программы в C?

a) Определение условий выполнения программы

b) Процесс компиляции программы

c\*) Включение или исключение частей кода при компиляции на основе условий

d) Запуск программы в режиме отладки

10. Как объявить макрос в C с использованием директивы #define?

a) macro x = 5;

b) define x(5);

c\*) #define x 5

d) const x = 5;

11. Как объявить глобальную переменную в C?

a\*) Вне всех функций, в начале файла

b) Внутри функции main()

c) Всегда локальная, нельзя объявить глобальную

d) С помощью ключевого слова global

12. Каким образом можно передать адрес переменной в функцию в C?

a) Через глобальные переменные

b) С помощью параметров функции

c\*) С помощью указателей

d) Через ключевое слово address

13. Как объявить статическую переменную внутри функции в C?

a) static int x;

b) int static x;

c\*) static int x = 5;

d) int x;

14. Какой оператор используется для возврата значения из функции в C?

a) break

b) continue

c\*) return

d) exit

Указатели и динамическая память. Выделение и освобождение динамической памяти. Средства Windows для работы с памятью. Передача параметров функций по ссылке и по указателю.

1. Что такое указатель в языке программирования C?

a) Число, указывающее на адрес в памяти

b) Специальная переменная для хранения текста

c\*) Переменная, которая содержит адрес в памяти другой переменной

d) Функция для выделения динамической памяти

2. Как объявить указатель на целое число в C?

a) int ptr;

b) ptr int;

c\*) int \*ptr;

d) pointer(int) ptr;

3. Какой оператор используется для получения значения, на которое указывает указатель в C?

a\*) \*

b) &

c) ->

d) .

4. Как выделить динамическую память под целочисленную переменную в C?

a) int dynamic = new int;

b) int\* dynamic = new int;

c) int dynamic = malloc(int);

d \*) int\* dynamic = malloc(int);

5. Как освободить динамическую память после использования в C?

a) delete dynamic;

b \*) free(dynamic);

c) delete[] dynamic;

d) destroy dynamic;

6. Что такое утечка памяти (memory leak) в программировании?

a) Ситуация, когда программа зависает

b) Ошибка компиляции

c) Некорректное завершение программы

d\*) Незавершенное освобождение выделенной памяти, что приводит к утрате доступа к части памяти

7. Какие функции используются для работы с динамической памятью в C?

a \*) malloc и free

b) allocate и deallocate

c) new и delete

d) create и destroy

8. Как можно передать указатель на функцию в качестве параметра в C++?

a) С помощью оператора "&"

b) Нельзя передавать указатель на функцию

c\*) С помощью имени функции без скобок

d) С помощью ключевого слова "function"

9. Что такое "ссылка" в C?

a) Адрес переменной

b) Тип переменной

c\*) Псевдоним переменной

d) Указатель на переменную

10. Как объявить ссылку на целое число в C?

a) int \*ref;

b) ref int;

c\*) int &ref;

d) reference(int) ref;

11. Какая из следующих операций невозможна при работе с указателями?

a) Увеличение значения указателя

b) Уменьшение значения указателя

c) Присвоение значения указателю

d\*) Сложение указателей

12. Какая из следующих функций используется для изменения размера динамически выделенной памяти в C?

a) resize()

b) realloc()

c) enlarge()

d\*) Нет такой функции

13. Как передать параметр по ссылке в C?

a) С помощью указателя

b) С помощью оператора "&"

c\*) Объявив функцию с амперсандом (&) после типа параметра

d) Объявив параметр с ключевым словом "reference"

14. Как передать параметр по указателю в C?

a) Объявив функцию с амперсандом (&) после типа параметра

b) Объявив параметр с ключевым словом "pointer"

c\*) Объявив параметр с типом указателя

d) С помощью оператора "\*"

15. Какой оператор используется для получения адреса переменной в C?

a\*) &

b) \*

c) ->

d) .

16. Какой оператор используется для доступа к членам структуры через указатель в C?

a) \*

b\*) ->

c) ::

d) .

17. Как объявить массив указателей на целые числа в C?

a) \*int array[5];

b) int array\*[5];

c\*) int \*array[5];

d) pointer(int) array[5];

Файлы. Текстовые и двоичные файлы, доступ к файлам в языке С. Средства Windows для работы с файлами.

1. Какой из следующих типов файлов содержит данные в виде текста?

a) Двоичный файл

b\*) Текстовый файл

c) Исполняемый файл

d) Системный файл

2. Каким образом можно открыть файл для чтения в C?

a) open("file.txt", "r");

b\*) FILE \*file = fopen("file.txt", "r");

c) read("file.txt");

d) ifstream file("file.txt");

3. Каким образом можно открыть файл для записи в C?

a\*) FILE \*file = fopen("file.txt", "w");

b) open("file.txt", "w");

c) write("file.txt");

d) ofstream file("file.txt");

4. Как закрыть файл после работы с ним в C?

a) fcloseFile("file.txt");

b) close("file.txt");

c\*) fclose(file);

d) exit();

5. Как проверить, успешно ли открыт файл в C?

a) check(file);

b\*) if (file != NULL)

c) open(file);

d) validate(file);

6. Как записать строку в текстовый файл в C?

a) file.writeLine("Hello");

b) file.output("Hello");

c\*) fprintf(file, "Hello");

d) file << "Hello";

7. Как прочитать строку из текстового файла в C?

a) file.readLine();

b) file.read("Hello");

c\*) fscanf(file, "%s", str);

d) getline(file, str);

8. Каким образом можно открыть двоичный файл для чтения и записи в C?

a\*) FILE \*file = fopen("binary.dat", "rb+");

b) open("binary.dat", "rb+");

c) ifstream file("binary.dat", ios::binary);

d) create("binary.dat", "rb+");

9. Какой режим открытия файла следует использовать для записи двоичных данных в C?

a) ios::in

b) ios::out

c\*) "wb" (для fopen)

d) ios::text

10. Каким образом можно проверить, достигнут ли конец файла в C?

a) file.endOfFile();

b) file.eof();

c\*) feof(file)

d) file.isEnd();

11. Каким образом можно переместить указатель текущей позиции внутри файла в C?

a) file.move(10);

b) file.jump(5);

c\*) fseek(file, 20, SEEK\_SET);

d) file.position(15);

РАБОТА СО СТРОКАМИ И ПАМЯТЬЮ

1. Какая библиотека используется для работы со строками в языке C?

a) iostream

b\*) string.h

c) cstring

d) cstring

2. Как объявить строку в C с использованием стандартной библиотеки?

a) string myString;

b) char\* myString;

c\*) char myString[100];

d) str myString;

3. Какой символ завершает строку (нулевой символ) в C?

a) \n

b) ;

c) \*

d\*) \0

4. Каким образом можно скопировать одну строку в другую в C?

a) copy(string1, string2);

b) strcopy(string1, string2);

c\*) strcpy(string1, string2);

d) copyString(string1, string2);

5. Какой функцией можно найти длину строки в C?

a) len(string);

b\*) strlen(string);

c) length(string);

d) size(string);

6. Каким образом можно объединить две строки в одну в C?

a) merge(string1, string2);

b) strcat(string1, string2);

c) combine(string1, string2);

d\*) strcat(string1, string2);

7. Каким образом можно сравнить две строки в C?

a) compare(string1, string2);

b) strcompare(string1, string2);

c) isEqual(string1, string2);

d\*) strcmp(string1, string2);

8. Какие функции используются для поиска подстроки в строке в C?

a) search(string, substring);

b) findSubstring(string, substring);

c\*) strstr(string, substring);

d) locateSubstring(string, substring);

9. Каким образом можно выделить подстроку из строки в C?

a) getSubstring(string, start, end);

b) extractSubstring(string, start, end);

c\*) strncpy(substring, string + start, end - start);

d) sliceSubstring(string, start, end);

10. Каким образом можно преобразовать строку в число (целое) в C?

a) convertToInteger(string);

b) toInt(string);

c\*) atoi(string);

d) stringToInteger(string);

11. Как создать копию строки в C++?

a) copy(string);

b) strcopy(string);

c\*) std::string copy = string;

d) std::string copy(string);

Работа с датами и временем. Функции с переменным количеством параметров. Битовые поля.

Конечно, вот 20 вопросов для теста по теме "Работа с датами и временем, функции с переменным количеством параметров, битовые поля":

Какое стандартное заголовочное имя в C/C++ используется для работы с датой и временем?

a) datetime.h

b\*) time.h

c) date.h

d) clock.h

Какое стандартное структуры данных используется для представления даты и времени в C/C++?

a) Date

b) Calendar

c\*) struct tm

d) TimeStamp

Как получить текущее время в C/C++?

a) getCurrentTime()

b) time.now()

c\*) time(NULL)

d) getTimeStamp()

Каким образом можно отформатировать дату и время в строку в C/C++?

a) formatDateTime(dateTime)

b\*) strftime(buffer, sizeof(buffer), "%Y-%m-%d %H:%M:%S", &timeInfo)

c) toString(dateTime)

d) formatDate(dateTime)

Как получить день недели для заданной даты в C/C++?

a) getWeekday(dateTime)

b) weekday(dateTime)

c\*) timeInfo.tm\_wday

d) dayOfWeek(dateTime)

Каким образом можно добавить определенное количество секунд к дате и времени в C/C++?

a) addSeconds(dateTime, seconds)

b) dateTime + seconds

c\*) mktime(&timeInfo)

d) increaseDateTime(dateTime, seconds)

Как объявить функцию с переменным количеством параметров в C/C++?

a) void myFunction(args...)

b) myFunction(args...)

c\*) void myFunction(...)

d) myFunction(...)

Как передать переменное количество аргументов в функцию в C/C++?

a) передать их в виде массива

b\*) использовать многоточечный аргумент "..."

c) использовать параметр "args"

d) передать их через указатель на структуру

Как получить аргументы с переменным количеством параметров внутри функции в C/C++?

a) использовать макросы VAR\_ARGS

b) преобразовать их в строку

c\*) использовать функции va\_start и va\_arg

d) передать их как структуру

Каким образом можно создать битовое поле в C/C++?

a) с помощью ключевого слова "bitfield"

b) использовать структуру и битовые операции

c\*) объявить структуру с битовыми полями

d) использовать ключевое слово "bit"

Какой тип данных используется для определения битового поля в C/C++?

a) int

b\*) unsigned int

c) char

d) float

Сколько битов выделяется под битовое поле типа unsigned int в следующем объявлении: unsigned int flag : 4;?

a) 1 бит

b) 2 бита

c\*) 4 бита

d) 8 битов

Каким образом можно установить значение битового поля в C/C++?

a) использовать функцию setBitfield()

b) присвоить значение с помощью оператора "="

c\*) использовать операторы битовых масок и сдвигов

d) вызвать функцию set()

Как объявить битовое поле с помощью структуры в C/C++?

a\*) struct BitFieldStruct { unsigned int flag1 : 1; unsigned int flag2 : 2; };

b) bitfield struct BitFieldStruct { flag1:1; flag2:2; };

c) struct BitFieldStruct { bit flag1 : 1; bit flag2 : 2; };

d) struct BitFieldStruct { bitfield flag1 : 1; bitfield flag2 : 2; };

Обработка исключительных ситуаций, сигналы. Нелокальные переходы.

Что такое исключение в контексте программирования на языке C?

a) Ошибка компиляции

b) Ошибка логики программы

c) Ошибка ввода-вывода

d\*) Необычная ситуация, которая может привести к некорректной работе программы

Каким образом можно сгенерировать исключение в языке C?

a) Используя оператор throw

b) Используя оператор catch

c) Используя оператор try

d\*) Используя функцию longjmp

Какой оператор используется для обработки исключений в языке C?

a) try-catch

b) if-else

c\*) setjmp-longjmp

d) for-in

Что произойдет, если исключение не будет поймано с помощью функции setjmp и longjmp?

a) Программа завершит выполнение без ошибок

b) Программа аварийно завершит выполнение

c\*) Программа продолжит выполнение сразу после вызова longjmp

d) Программа продолжит выполнение сразу после вызова setjmp

Какой оператор используется для завершения выполнения текущей функции и генерации исключения в языке C?

a) throw

b\*) longjmp

c) return

d) exit

Как можно передать информацию о произошедшем исключении в языке C?

a) С помощью оператора return

b) С помощью оператора throw

c\*) С помощью параметра функции setjmp

d) С помощью глобальной переменной

Что такое сигнал в контексте операционной системы?

a\*) Сообщение, которое уведомляет программу о возникновении определенного события или ошибки

b) Функция в C для обработки ошибок

c) Событие, вызванное пользователем

d) Способ взаимодействия программы с операционной системой

Каким образом можно обработать сигнал в языке C?

a\*) С помощью функции signal

b) С помощью оператора try-catch

c) С помощью оператора if-else

d) С помощью ключевого слова catch

Что такое нелокальный переход (longjmp) в языке C?

a) Принудительное завершение программы

b) Возврат к предыдущей точке выполнения программы

c\*) Переход к определенной точке программы, которая не является непосредственно вызывающей функцией

d) Остановка программы

Как можно выполнить нелокальный переход с использованием функции longjmp?

a) С помощью оператора return

b) С помощью оператора throw

c\*) Вызвать функцию longjmp с указанием метки перехода

d) С помощью оператора goto

Какие аргументы принимает функция setjmp в языке C?

a) Ни одного аргумента

b) Один аргумент типа int

c\*) Один аргумент типа jmp\_buf

d) Два аргумента типа int

Какие аргументы принимает функция longjmp в языке C?

a) Ни одного аргумента

b) Один аргумент типа int

c\*) Два аргумента: объект jmp\_buf и значение int

d) Два аргумента: указатель на функцию и значение int

Каким образом можно использовать нелокальные переходы в языке C?

a) Для обработки исключений

b) Для перехода между функциями

c\*) Для реализации сложной логики управления программой в нестандартных ситуациях

d) Для создания глобальных переменных