1. Перечислите фундаментальные типы данных Си?

2. Какие есть квалификаторы для целых типов данных? (long, short, signed и unsigned)

3. Какие есть квалификаторы для числе с плавающей точкой? (long)

4. Какими способами можно задать константу в Си?

5. Какие Escape-последовательности вы можете назвать?

6. Как и с какого стандарта можно использовать тип bool?

7. Какая функция используется для вывода данных в консоль для ввода данных с консоли?

8. Какое расширение у файла программы на языке Си?

9. Как скомпилировать написанную на языке Си программу?

10. Какие есть виды операций в Си?

11. Как работать с перечислениями в языке Си?

12. Какие условные операторы есть в языке Си?

13. Какие операторы цикла есть в языке Си?

14. В чем разница между операторами перехода break и continue?

15. Какая функция используется для генерации псевдослучайных чисел? В каком диапазоне значений они генерируются?

16. Назовите формулу для генерации псевдослучайного числа на интервале [a, b].

17. Какая функция устанавливает начальное значение для генерации псевдослучайного числа?

18. Какие есть способы инициализации статических массивов данных?

19. Что такое указатель.

20. Арифметика указателей.

21. Функции работы с динамической памятью?

22. Поясните понятие «С-строка».

23. В чем отличие статического и динамического массива?

24. Области видимости переменных.

25. Передача массивов в качестве параметров.

26. Описание функции с переменным числом параметров.

27. Дайте определение структуры.

28. Дайте определение директивы препроцессора. Перечислите использованные вами директивы.

29. Опишите принцип работы с файлами в Си.

30. Дайте писание функции main().

Задачи для проведения зачета

1. Напишите функцию, которая принимает массив с разными числами, а возвращает максимальное произведение двух чисел из этого массива.

Входные данные:

[1, 3, 2, 2, 3, 0]

[-8, 1, 2, -3, 0, 4]

[5, 6, 0, -1, 2]

[10, -2, -6, 5, 11]

[9, 0, 9, 1, -9]

Выходные данные:

9

24

30

110

81

2. Напишите функцию, которая принимает строку, а возвращает ее перевернутый вариант (слова в обратном порядке).

Входные данные:

"hello world"

“general purpose programming language”

Выходные данные:

"world hello"

“language programming purpose general”

3. На вход подается число N. Вывести на экран треугольник из символов ’\*’ из N линий.

Пример для трёх линий:

\*

\*\*\*

\*\*\*\*\*

4. Реализовать функцию сортировки массива.

Входные данные:

[3, 4, 9, -1, 0, 10, 5]

Выходные данные:

[-1, 0, 3, 4, 5, 9, 10]

5. Дано квадратное уравнение вида: a\*x^2 + b\*x + c = 0

Программа считывает вещественные коэффициенты a, b, c и выводит решения уравнения. Если корней нет, выводит сообщение об этом.

6. Игра "Угадай число". Программа "загадывает" (генерирует) случайное число k в диапазоне от 1 до 100.

Пользователь вводит числа в программу и получает ответ вида "слишком много" или "слишком мало" (по сравнению с "загаданным" числом). Как только пользователь вводит в программу "загаданное" число k, выводится сообщение о победе и программа завершается.

7. На вход подаются коэффициенты a\_0, a\_1, ..., a\_n многочлена n-й степени. Найти значение многочлена в точке, введенной пользователем.

8. Реализовать функции работы со строками, такие как: преобразование к верхнему регистру и наоборот, подсчет нескольких заданных символов в строке, удаление заданного символа из строки, проверка является ли строка палиндромом, замены заданного символа.

9. На вход подается одномерный массив целых чисел (либо можно сделать через ввод N и генерацию N чисел в массив, при генерации мы можем ограничить [a,b]). Необходимо отсортировать массив (можно методом пузырька) и вывести его на экран. В отсортированном массиве найти медиану (отметка, делящая ранжированные данные, либо число по середине ранжированного списка либо среднее арифметическое из двух по середине), моду (наиболее часто встречающееся значение в данных), среднее, максимум, минимум, размах (разность между наибольшим и наименьшим).

10. Комплексные числа. Реализовать модуль для работы с комплексными числами.

Модуль должен содержать следующие функции и структуры:

// Комплексное число

typedef struct

{

float x;

float y;

} Complex;

// Сложение

Complex add(Complex a, Complex b);

// Умножение

Complex mul(Complex a, Complex b);

// Деление

Complex div(Complex a, Complex b);

// Получение модуля

float mod(Complex a);

// Получение аргумента

float add(Complex a);

// Получение сопряжённого

Complex add(Complex a);

// Вывод в алгебраической форме

void printAlg(Complex a);

// Вывод в тригонометрической форме

void printTrig(Complex a);

// Вывод в показательной форме

void printExp(Complex a)

11. В файле записана информация о зачете в текстовом виде (фамилии латиницей в транслите, unix перевод строки). Информация о студенте описывается структурой:

typedef struct

{

char lastName[64]; // фамилия

int id; // шифр

int mark; // оценка: 0 - fail, 1 - pass

} Student;

− Название файла передается первым параметром командной строки.

− В файле первой строкой идет количество студентов.

− В последующих:

− фамилия, записанная наоборот

− шифр

− оценка: `0` или `1`

Например:

vogedanzeB

133742

0

1) Сделать функцию разворачивания строки void reverse(char\* string).

2) Сделать функцию вывода void print(const Student\* student), которая выводит информацию об одном студенте в формате:

LastName: фамилия (в правильном виде)

Id: шифр

Mark: оценка (в виде слова. Если 0, то fail. Если 1, то pass)

3) Вывести информацию о студентах в stdout. Фамилия должна быть в правильном (развернутом) виде. Оценка в виде слова (`fail`, `pass`).

4) В конце распечатать процент сдавших в формате:

Passed: процент сдавших

5) При успешном выполнении программы в качестве кода возврата возвращать `0`. В случае отсутствия аргументов командной строки, отсутствия файла, ошибки чтения файла (количество данных меньше чем указано в первой строке файла) в качестве кода возврата возвращать `1`.

Пример ввода:

2

vonavI

156256

1

vogedanzeB

133742

0

Пример вывода:

LastName: Ivanov

Id: 156256

Mark: pass

LastName: Beznadegov

Id: 133742

Mark: fail

Passed: 50.000000