

Язык программирования Си

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

Семинар 5. Коллоквиум, методика проведения и типовые вопросы.

Коллоквиум проводится на семинарском занятии в середине семестра (обычно – до 1-го ноября) в форме тотального опроса с билетами. На семинаре, следующим за коллоквиумом, рекомендуется провести разбор типичных ошибок студентов при ответе на вопросы.

1. Билеты

1. Билет

- | | |
|--|---|
| 1) Основы синтаксиса языка Си. Ключевые слова. Фундаментальные типы данных. Определение переменных и констант. Выражения, операции, комментарии. | 2) Разные виды цикла <code>while</code> , <code>do</code> . |
|--|---|
-

2. Билет

- | | |
|--|--|
| 1) Операторы в выражениях языка Си. Приоритет операторов. Оператор <code>sizeof()</code> . | 2) Директива <code>#define</code> препроцессора и ее использование. Макроопределения с параметром. |
|--|--|
-

3. Билет

- | | |
|---|--|
| 1) Управляющий оператор <code>switch</code> . | 2) Директивы условной компиляции препроцессора и их использование. |
|---|--|
-

4. Билет

- | | |
|--|---|
| 1) Управляющие операторы <code>if</code> , <code>goto</code> . | 2) Модульный подход в программировании. Использование <code>*.h</code> файлов. Раздельная компиляция. |
|--|---|
-

5. Билет

- | | |
|--|---|
| 1) Массивы. Передача массивов в функции. | 2) Локальные, глобальные, статические переменные. |
|--|---|
-

6. Билет

- | | |
|---|----------------------------|
| 1) Основные действия, выполняемые при помощи отладчика в среде <i>Microsoft Visual Studio</i> . | 2) Цикл <code>for</code> . |
|---|----------------------------|
-

7. Билет

- | | |
|--|--|
| 1) Функция <code>main()</code> , ее параметры. | 2) Оператор <code>typedef</code> . Приведение типов. |
|--|--|
-

8. Билет

- | | |
|--|--|
| 1) Использование функций: заголовок, тело и вызов функции. | 2) Управляющие операторы <code>if, goto</code> . |
|--|--|
-

9. Билет

- | | |
|-------------------------|--|
| 1) Статические функции. | 2) Блоки и правила видимости переменных. |
|-------------------------|--|
-

10. Билет

- | | |
|---|--|
| 1) Передача параметров в функции.
Передача параметров по значению. | 2) Операторы в выражениях языка Си.
Приоритет операторов.
Оператор <code>sizeof()</code> . |
|---|--|
-

11. Билет

- | | |
|---|----------------------------|
| 1) Работа в среде <i>Microsoft Visual Studio</i> : создание проекта. Основные файлы проекта. Этапы создания программы: компиляция, линковка, запуск на выполнение, отладка. | 2) Цикл <code>for</code> . |
|---|----------------------------|
-

12. Билет

- | | |
|---|--|
| 1) Разные виды цикла <code>while, do</code> . | 2) Массивы. Передача массивов в функции. |
|---|--|
-

13. Билет

- | | |
|--|--|
| 1) Директива <code>#include</code> препроцессора и ее использование. | 2) Функция <code>main()</code> , ее параметры. |
|--|--|
-

14. Билет

- | | |
|--|---------------------------------------|
| 1) Процедурный подход программирования. Определение функции. Прототип функции. | 2) Операторы инкремента и декремента. |
|--|---------------------------------------|
-

15. Билет

- | | |
|---|--|
| 1) Передача параметров в функции.
Передача параметров по значению. | 2) Директивы условной компиляции препроцессора и их использование. |
|---|--|
-

16. Билет

- | | |
|---|--|
| 1) Локальные, глобальные, статические переменные. | 2) Использование функций: заголовок, тело и вызов функции. |
|---|--|
-

17. Билет

- | | |
|--|--|
| 1) Директива <code>#define</code> препроцессора и ее использование. Макроопределения с параметром. | 2) Управляющий операторы <code>switch</code> . |
|--|--|
-

18. Билет

1) Разные виды цикла `while`, `do`.

2) Директива `#include` препроцессора и ее использование.

19. Билет

1) Работа в среде *Microsoft Visual Studio*: создание проекта. Основные файлы проекта. Этапы создания программы: компиляция, линковка, запуск на выполнение, отладка.

2) Глобальные и внешние переменные.

20. Билет

1) Виды операторов присваивания в языке Си.

2) Оператор `typedef`. Приведение типов.

21. Билет

1) Операторы инкремента и декремента.

2) Рекурсивный вызов функций.

22. Билет

1) Операторы в выражениях языка Си. Приоритет операторов. Оператор `sizeof()`.

2) Статические переменные.

23. Билет

1) Основные действия, выполняемые при помощи отладчика в среде *Microsoft Visual Studio*.

2) Массивы: одномерные и двумерные.

24. Билет

1) Разные виды цикла `while`, `do`.

2) Модульный подход в программировании. Использование `*.h` файлов. Раздельная компиляция.

25. Билет

1) Логические (булевские) операторы и операторы сравнения.

2) Цикл `for`.

2. Факультативные задания

Так как коллоквиум проводится на семинарском занятии, то у студентов появляются дополнительные 2 недели, которые следует использовать для доделки и сдачи всех выданных ранее задач. Практика показывает, что к этому моменту «хвосты» имеет, как минимум, половина группы.

Студенты, которые успешно сдали все предыдущие задания, могут получить факультативное задание. Учет этих факультативных задач отличается от учета обычных заданий: они не являются обязательными, то есть, отсутствие их решения – не ухудшает оценки за успеваемость студента в течение семестра, в то же время, наличие такой сданной задачи улучшает оценку студента. Если студент не успевает их сдать до 6-го семинара, то такому студенту рекомендуется сначала сдавать плановые, обычные задания и лишь когда плановые задания сданы – вернуться к выполнению факультативных заданий.

Во всех приведенных ниже заданиях нужно оформить основную часть задания в виде функции, которой передаются все необходимые аргументы и которая возвращает результат. Весь ввод-вывод сосредоточить в функции `main()`.

1. Вариант

Написать калькулятор, выполняющий арифметические операции с шестнадцатеричными числами.

Шестнадцатеричные числа – произвольной разрядности, представляются десятичными цифрами 0-9 и буквами A-F, соответствующими числам с 10 до 15. У пользователя запрашивается пара чисел, печатается ответ. Преобразование чисел из текстового шестнадцатеричного вида в целочисленный выполнить в отдельной функции самостоятельно. Для печати ответа воспользоваться оператором:

```
cout << hex << result << endl << flush;
```

2. Вариант

Написать программу, вычисляющую определитель квадратной матрицы произвольной размерности.

Размерность и коэффициенты матрицы (не более 10) запрашиваются у пользователя, программа должна напечатать матрицу и ее определитель.

3. Вариант

Написать программу, выполняющую символьное умножение «в столбик» двух шестнадцатеричных чисел, вводимых пользователем «цифра за цифрой».

В программе использовать таблицу умножения для шестнадцатеричных цифр и правила умножения «в столбик». Не использовать приведение шестнадцатеричного числа к десятичной системе счисления.

4. Вариант

Написать программу, которая решает систему линейных уравнений (не выше 6-го порядка) методом Гаусса (приведением матрицы к треугольному виду). Для отладки и тестирования можно использовать следующую систему:

$$\begin{aligned}x + y - z + t &= 4 \\2x - y + 3z - 2t &= 1 \\x - y + 2t &= 6 \\3x - y + z - t &= 0\end{aligned}$$

Для систем уравнений не имеющих решения печатать сообщение об ошибке.

5. Вариант

Написать и протестировать программу, выполняющую поиск подстроки в тексте методом Боуера-Мура (сравнивая символы, начиная с конца подстроки и учитывая, что символы текста, отсутствующие в искомой подстроке должны приводить к сдвигу подстроки по тексту за текущий символ).

6. Вариант

Отсортировать массив из уникальных элементов алгоритмом QuickSort: выбрать случайный элемент из середины массива, двигаясь от начала массива остановиться на элементе, который больше выбранного, двигаясь от конца массива остановиться на элементе, который меньше выбранного. Обменять найденные два элемента. Продолжить движение с обоих концов массива к его центру. Получив разбиение массива, в котором все элементы до некоторого – меньше выбранного, а все элементы после него – больше выбранного, повторить алгоритм рекурсивно для каждой из двух частей разбиения.

Сравнить количество операций сравнения и присваивания элементов массива с аналогичными показателями пузырьковой сортировки.

7. Вариант

Вычислить число π методом Монте-Карло: взять круг, вписанный в квадрат со стороной 10. Сгенерировать 1000 пар случайных чисел, каждое в диапазоне от 0 до 10. Каждую пару чисел считать x - и y - координатой некоторой точки внутри заданного квадрата. Для каждой такой точки определить – лежит ли она внутри круга? Отношение числа точек, попавших в круг к общему числу точек оценивает величину $\pi / 4$.

Сравнивая результат со встроенной константой M_PI оценить погрешность этого метода в зависимости от выбранного количества пар случайных чисел.

8. Вариант

Построить свой собственный генератор случайных чисел, основанный на линейном конгруэнтном методе: $x_{i+1} = (ax_i + c) \% M$ ($a < M$, $c < M$). Построить (в виде символьной матрицы 20×20) гистограмму достаточно большой сгенерированной последовательности чисел.

По гистограмме «на глаз» оценить равномерность распределения в зависимости от разных значений параметров a , c , M .

9. Вариант

Методом Эйлера решить задачу о движении снаряда, запущенного под углом к горизонту с учетом силы сопротивления воздуха, пропорциональной скорости его движения.

Программа должна принимать начальные параметры: угол в градусах, модуль начальной скорости и коэффициент сопротивления воздуха и печатать ответ: расстояние до точки падения снаряда.

10. Вариант

Написать рекурсивную версию алгоритма, переводящего заданное число в двоичную систему счисления.

11. Вариант

Написать рекурсивную версию алгоритма, распечатывающего все простые множители заданного числа.
