

ЛАБОРАТОРНАЯ РАБОТА №1

Цель работы:

приобрести умения и практические навыки для работы по созданию линейных программ и программ с разветвлениями; приобрести умения и практические навыки для работы по созданию программ с циклами.

Теоретическая часть:

Алфавит – основные неделимые знаки, с помощью которых пишутся все тексты программы (0...9, a...z, A...Z, специальные знаки – (), [], {}, ‘, “”, +, -, *, /, %, точка, запятая, двоеточие, точка с запятой, ~, ^, !, ?, #, &, ||, пробел).

Пробельные символы – знак пробел, знак табуляции, знак перехода на новую строку.

Из символов алфавита формируются *лексемы* – минимальные единицы языка, имеющие самостоятельный смысл. К ним относятся: идентификаторы, ключевые слова, знаки операций, константы, разделители. Границы лексем определяются другими лексемами.

Идентификатор – имя программных объектов. Идентификаторы состоят из букв, цифр и знака нижнего подчеркивания. Они не могут начинаться с цифры, не допускаются пробелы, теоретически их длина не ограничена. Нельзя начинать их с символа нижнего подчеркивания, а также нельзя выбирать в качестве идентификатора *ключевые слова* – зарезервированные идентификаторы, имеющие специальное назначение. Различаются строчные и заглавные символы. Идентификаторы могут быть, например, такими: *A1*, *Kol_vo*, *x*.

Знаки операции – один или более символ, которые определяют действие над операндом. Знаки пробела не допустимы.

Константы бывают следующих типов:

– *целые* – десятичные, восьмеричные (начинаются с 0), шестнадцатеричные (начинаются с 0x или 0X);

– *вещественные* – с десятичной точкой, экспоненциальная форма записи – $8e^{-13} = 8 * 10^{-13}$, $1.3E126 = 1.3 * 10^{126}$;

– *символьные* – один или более символов, заключенные в апострофы (‘а’, ‘*’, ‘ф’). *Escape-последовательность* – последовательность, которая начинается с обратной косой черты. Управляющая последовательность, которая рассматривается как один символ (‘\n’, ‘\t’, ‘\’ и т.д.);

– *строковые* – последовательность символов, заключенная в кавычки (“а”, “строка”). Если идут несколько строк подряд, они объединяются в одну. Длинную строковую константу можно записать на нескольких строчках, при этом обратная косая черта будет обозначать знак переноса.

Комментарий при компиляции кода не учитывается. Если комментарий располагается в одну строчку, то он должен начинаться с //. Если он состоит из нескольких строк, то синтаксис комментария таков: /* комментарий */, например:

```
/* это комментарий,  
   который располагается  
   на трех строчках */
```

Как правило, программа на языке Си состоит из следующих элементов:

- директивы препроцессора;
- описание;
- функции.

Директивы препроцессора – директивы, используемые для присоединения специальных файлов, содержащих описание применяемых в программе функций.

Описание – в данном блоке описываются переменные и типы, используемые в программе.

Функции – любая программа состоит из них – могут вызывать друг друга, но не могут содержаться одна в другой. Одна из функций обязательно имеет имя *main* – с нее начинается выполнение программы. Существует несколько форматов этой функции, один из которых выглядит следующим образом:

```
void main () {тело функции}
```

Тело функции содержит описания и операторы языка Си.

Тип данных определяет:

- внутреннее представление данных в памяти;
- множество значений, которые могут принимать переменные этого типа;
- операции, которые можно применять к переменным данного типа.

Типы подразделяют на *основные* и *составные*.

Основные типы – арифметические типы – типы, которые можно использовать в арифметических операциях. Существуют следующие арифметические типы:

- *целый* – *int* – можно перед данным типом также писать *short* (тогда размер составит 2 байта) или *long* (4 байта). Целое число в двоичной системе счисления во внутреннем представлении. Если *signed* (по умолчанию) – со знаком, *unsigned* – без знака;

- *символьный* – *char* – под него отводится количество байт, необходимое для представления любого символа данного компьютера (1 байт для латиницы). Также может быть *signed* (по умолчанию) или *unsigned*. Данный тип может использоваться для представления целых чисел от -128 до 127 или от 0 до 255;

- *вещественный* – *float* (одинарной точности), *double* (удвоенной точности), *long double* (максимальной точности);

- тип *void* – тип, множество значений которого пусто. Используется для описания функций, которые не возвращают результаты в вызывающую программу и указания пустого списка параметров.

Операторы делятся на *исполняемые* и *неисполняемые*.

Исполняемые операторы задают действие над данными. *Неисполняемые* являются описаниями данных (описания). Каждый оператор завершается *точкой с запятой*.

Переменные – поименованная область памяти, в которой хранятся данные определенного типа. Имеют имя и значение. Бывают *локальные* и *глобальные*.

Локальные переменные объявлены внутри блока, их область действия – от точки объявления и до конца блока (включая внутренние блоки).

Глобальные переменные объявлены вне блока, действуют от точки описания и до конца файла. Любая переменная должна быть описана перед использованием.

Оператор описания переменной имеет следующий синтаксис:

```
класс const тип имя [размерность] = инициализатор;
```

- *имя* – правильный идентификатор;
- *тип* – любой тип в языке Си;
- *инициализатор* – позволяет присвоить переменной начальное значение при ее объявлении;
- *const* – указывает, что переменную нельзя изменять. Поименованная константа. Должна быть инициализирована при объявлении;
- *размерность* – необходима для описания массивов;
- *класс памяти* определяет время жизни и область видимости программного объекта.

Можно объявлять несколько переменных, разделяя их знаком запятой.

Выражения состоят из операндов и знаков операций. Они используются для вычисления значения определенного типа.

Операции бывают *унарные* (один операнд), *бинарные* (два операнда) и *тернарные* (три операнда). Каждый операнд может быть выражением или, в частном случае, константой или переменной. Если рядом расположены операции одного приоритета, то:

- справа налево вычисляются унарные, тернарная (условная), операция присваивания;
- слева направо вычисляются все остальные операции.

Операции *инкремента* (++) и *декремента* (--) имеют две формы записи:

- *префиксная* – операнд изменяется, новое значение является результатом выражения;

```
int a = 2, b;  
b = ++a; // a = 3, b = 3
```

– *постфиксная* – результат выражения – исходное значение операнда, после вычисления выражения операнд изменяется.

```
b = a++; // b = 2, a = 3
```

Результатом выполнения *операций сравнения* является значение «истина» или «ложь». Существуют следующие операции сравнения: ==, !=, >, >=, <, <=).

Логические операции – «и» (&&), «или» (||) – выполняются над всем аргументом в соответствии с таблицей истинности каждой из операций.

Операция *отрицания* в языке Си выглядит следующим образом:

```
int x = 5, y;  
y = -x; // y = -5  
y = !x; // y = 0
```

Операция *деления и остатка от деления (%)* применяется только к целым, результат также будет целым.

Преобразование типа (type) производится следующим образом:

```
a = x / (float) y; // a = 3,5 при x = 7 и y = 2
```

Результат вычисления выражения имеет значение и тип. Если операнды одного типа, то результат имеет тот же тип. Если операнды имеют различный тип, то перед вычислением выполняется преобразование от более коротких к более длинным типам для сохранения значимости и точности, при этом может измениться внутреннее представление переменных.

Приоритет	Действие
Унарные операции	
2	++ увеличить на 1 -- уменьшить на 1
2	~ поразрядное отрицание ! логическое отрицание - арифметическое отрицание
2	+ унарный плюс
2	& взятие адреса
2	* разыменование

Приоритет	Действие
2	<i>(type)</i> преобразование типа
2	<i>sizeof</i> размер
Бинарные и тернарные операции	
3	* умножение / деление % остаток от деления
4	+ сложение - вычитание
5	<< сдвиг влево >> сдвиг вправо
6	< меньше <= меньше или равно > больше >= больше или равно
7	= = равно != не равно
8	& поразрядная конъюнкция И
9	^ поразрядное исключающее ИЛИ
10	поразрядная дизъюнкция
11	&& логическое И
12	логическое ИЛИ
13	? : условная операция (тернарная)
14	= присваивание * = умножение с присваиванием / = деление с присваиванием % = остаток от деления с присваиванием + = сложение с присваиванием - = вычитание с присваиванием

Приоритет	Действие
	<< = сдвиг влево с присваиванием >> = сдвиг вправо с присваиванием & = поразрядное И с присваиванием = поразрядное ИЛИ с присваиванием ^ = поразрядное искл. ИЛИ с присваиванием
15	, последовательное вычисление

В языке Си существуют различные операторы.

выражение; – оператор вычисления выражений, частный случай выражения – пустой оператор, он используется, когда по синтаксису требуется оператор, а по смыслу – нет.

Операторы ветвлений – *условный оператор if (выражение) оператор1; else оператор2;* – выражение должно иметь арифметический тип или тип указатель. Если нужно выполнить несколько операторов, то их заключают в фигурные скобки. Типичной ошибкой при использовании операторов ветвления является неправильное использование операций сравнения, например, использование операции присваивания вместо операции сравнения и т.д., а также запятая с точкой, поставленная после оператора *if* ().

Также к операторам ветвления относится *оператор ветвления switch*. Он имеет следующий синтаксис:

```
switch (выражение) {
  case константное выражение 1: оператор1;
  ...
  case константное выражение N: оператор N
  default: оператор; }
```

Выражение должно быть целочисленным. Данный оператор предназначен для разветвления процесса вычисления на несколько направлений. Выполнение оператора начинается с вычисления выражения, результат сравнивается с константными выражениями. Управление передается первому оператору из списка, помеченного константным

выражением, совпавшим с результатом выражения. Затем выполняются операторы из всех ветвей, включая *default* – необязательную ветвь, которая получает управление, если не одно константное выражение не совпало с результатом.

Оператор *break* применяется для того, чтобы избежать выполнение операторов и других ветвлений. Предназначен для выхода из оператора *switch* или операторов цикла. Этот оператор передает управление оператору, следующему за оператором *switch* или циклом.

Цикл – многократное выполнение одного или группы операторов. *Тело цикла* – выполняемые операторы. Однократное выполнение тела называется *итерацией*.

Существуют различные операторы цикла. Рассмотрим их далее.

```
while (выражение) оператор;
```

Этот оператор называется *оператором цикла с предусловием*. Если выражение не выполняется, происходит выход из цикла. Выражение должно быть арифметическим.

```
#include "stdafx.h"
void main() {
    int x;
    scanf ("%d", x);
    while (x) {
        printf ("%d", x);
        scanf ("%d", &x);}
}
```

или:

```
while (1) {
    scanf ("%d", &x);
    if (x) break;
    printf ("%d", x);}
}
```

Тело цикла при использовании этого оператора может ни разу не выполниться.

```
do оператор; while (выражение)
```

Этот оператор называется *оператором цикла с постусловием*. Выражение также должно быть арифметическим.

```
do {
```



```
scanf ("%d", &x);
printf ("%d", x);
} while (x);

for (инициализация; выражение; модификация) оператор;
```

Этот оператор цикла называется *оператором цикла с параметром*.

Инициализация – объявление и присвоение начальных значений величинам в цикле. Выражений может быть несколько. *Модификацией* называется изменение параметров цикла после каждой итерации. Инициализация производится один раз в начале цикла, причем переменные из нее действуют только в самом цикле. Любая часть может быть пустой, но точки с запятой сохраняются.

Оператор *continue*; передает управление на начало следующей итерации, пропуская все операторы до конца тела цикла.

Если в теле одного цикла содержится другой оператор цикла, то говорят о *вложенных циклах*.

Варианты заданий.

Номер варианта по списку	Вариант в лабораторной (1 задание)	Вариант в лабораторной (2 задание)
1	7	22
2	6	21
3	5	20
4	4	19
5	3	18
6	2	17
7	1	16
8	15	15
9	14	14
10	13	13
11	12	12
12	11	11
13	10	10

14	9	9
15	8	8
16	7	7
17	6	6
18	5	5
19	4	4
20	3	3
21	2	2
22	1	1

Задание 1.

Варианты:

1. Дано 4-значное целое число. Вычислить и вывести на экран сумму цифр этого числа. Число вводится с клавиатуры.
2. Дано 4-значное целое число. Определить, равна ли сумма старших цифр сумме младших. Число вводится с клавиатуры.
3. Определить число, полученное выписыванием в обратном порядке цифр заданного трехзначного числа, и вывести его на экран. Число вводится с клавиатуры.
4. Найти и вывести на экран произведение цифр заданного четырехзначного числа. Число вводится с клавиатуры.
5. Ввести с клавиатуры значение x , вычислить и вывести на экран значение y :

$$y = \begin{cases} x, & x \leq 0 \\ x^2, & 0 < x \leq 1 \\ x^3, & x > 1 \end{cases}$$

6. Ввести с клавиатуры значение x , вычислить и вывести на экран значение y :

$$y = \begin{cases} x, & x > 1 \\ x^4, & 0 < x \leq 1 \\ x^2, & x \leq 0 \end{cases}$$

7. Ввести число x – возраст человека. Определить: ходит ли он в ясли (от 1 до 3 лет), ходит ли он в детский сад (от 4 до 6 лет), ходит ли он в школу (от 7 до 18 лет), уже окончил школу (19 лет и старше).

8. Ввести число t от 0 до 24 – час дня. Определить, какому времени суток соответствует этот час: от 0 до 3 – ночь, от 4 до 10 – утро, от 11 до 17 – день, от 18 до 21 – вечер, от 22 до 24 – ночь. Число вводится с клавиатуры.

9. Дано: число t – температура воздуха. Выводить на экран сообщение о погоде: холодно, прохладно, тепло, жарко. Число вводится с клавиатуры.

10. Дано: a , b и c – углы треугольника в градусах. Определить, можно ли построить такой треугольник, и если можно, то определить тип треугольника. Значения углов вводятся с клавиатуры.

11. Дано: a , b и c – стороны треугольника. Проверить, можно ли построить треугольник с такими сторонами, и если можно, то определить его тип (равносторонний, равнобедренный, разносторонний). Значения сторон треугольника вводятся с клавиатуры.

12. Даны 4 числа: оценки студента на экзаменах. Если он получил хотя бы одну оценку «2» – его отчислят. Если он получил хотя бы одну оценку «3» – его лишат стипендии. Если он получил все оценки «5» – он получит повышенную стипендию. Вывести на экран сообщение о решении деканата. Оценки вводятся с клавиатуры.

13. Дано 3 целых числа. Вывести на экран «Да» или «Нет» в зависимости от того, имеют 3 числа одинаковую четность или нет. Числа вводятся с клавиатуры.

14. Даны целые числа a , b , c и d . Найти и вывести на экран наибольшее из них. Числа вводятся с клавиатуры

15. Даны целые числа a , b , c и d . Найти и вывести на экран число, которое ближе всего к среднему арифметическому этих чисел. Числа вводятся с клавиатуры.

Задание 2.

Варианты:

1. Вычислить и вывести на экран значения функции:

$$y = \begin{cases} 3x^2 + 2, & \text{если } 1 \leq x < 2 \\ x, & \text{если } 2 \leq x < 4 \\ \frac{5}{x-8}, & \text{если } 4 \leq x \leq 9 \end{cases}$$

при x , который изменяется на интервале от 1 до 9 с шагом 1.

2. Найти на интервале от 9,1 до 10 корень уравнения $x^3 + x = 1000$ с точностью до 0,0001.

3. Дана непустая последовательность положительных целых чисел, за которыми следует 0 (признак конца последовательности). Вычислить среднее арифметическое этих чисел. Последовательность должна вводиться с клавиатуры.

4. Дана непустая последовательность положительных целых чисел, за которыми следует 0 (признак конца последовательности). Вычислить сумму этих чисел. Последовательность должна вводиться с клавиатуры.

5. Дана непустая последовательность положительных целых чисел, за которыми следует 0 (признак конца последовательности). Вычислить произведение этих чисел. Последовательность должна вводиться с клавиатуры.

6. Вывести на экран латинские буквы, коды которых четны.

7. Вывести на экран латинские буквы, коды которых нечетны.

8. Вывести на экран латинские буквы, коды которых кратны введенному с клавиатуры числу от 1 до 10.

9. Дана дата в виде: день, месяц. Напечатать, сколько дней прошло от начала года. Дата вводиться с клавиатуры.

10. Дано целое $n > 0$, за которым следует n вещественных чисел. Определить, сколько среди них отрицательных. Значение n и вещественные числа вводятся с клавиатуры.

11. Дано целое $n > 0$, за которым следует n вещественных чисел. Определить, сколько среди них положительных. Значение n и вещественные числа вводятся с клавиатуры.

12. Дано целое $n > 0$, за которым следует n вещественных чисел. Определить, сколько среди них кратных 2. Значение n и вещественные числа вводятся с клавиатуры.

13. Дано целое $n > 0$, за которым следует n вещественных чисел. Определить, сколько среди них кратных числу, введенному с клавиатуры. Значение n и вещественные числа вводятся с клавиатуры.

14. Дана последовательность из N вещественных чисел. Определить, каких значений больше – положительных или отрицательных. Значение N и числа вводятся с клавиатуры.

15. Дана последовательность из N целых чисел. Найти среднее арифметическое максимального и минимального элементов последовательности. Значение N и числа вводятся с клавиатуры.

16. Дана последовательность из N целых чисел. Определить, сколько элементов последовательности находится в интервале от -1 до 45. Значение N и числа вводятся с клавиатуры.

17. Найти значения 10 членов ряда $1; \frac{1}{2}; \frac{1}{4}; \frac{1}{6}; \frac{1}{8}; \dots$.

18. Найти значения 40 членов ряда $1; \frac{3}{4}; \frac{4}{8}; \frac{5}{16}; \dots$.

19. Дана последовательность из n целых чисел. Определить, является ли она невозрастающей. Значение n и числа вводятся с клавиатуры.

20. Дана последовательность из n целых чисел. Определить, является ли она неубывающей. Значение n и числа вводятся с клавиатуры.

21. Дана последовательность из целых чисел, которая заканчивается нулем. Определить, является ли она невозрастающей. Числа вводятся с клавиатуры.

22. Дана последовательность из целых чисел, которая заканчивается нулем. Определить, является ли она неубывающей. Числа вводятся с клавиатуры.

23. Дана последовательность из n целых чисел. Определить, является ли она возрастающей. Значение n и числа вводятся с клавиатуры.

24. Дана последовательность из n целых чисел. Определить, является ли она убывающей. Значение n и числа вводятся с клавиатуры.

25. Дана последовательность из целых чисел, которая заканчивается нулем. Определить, является ли она возрастающей. Числа вводятся с клавиатуры.

26. Дана последовательность из целых чисел, которая заканчивается нулем. Определить, является ли она убывающей. Числа вводятся с клавиатуры.

27. Дано: x – целое число. Определить, является ли оно простым. Число вводится с клавиатуры.

28. Для каждого целого числа из интервала a до b , вывести на экран все его делители. Числа a и b вводятся с клавиатуры.

29. Дано целое число x . Вывести на экран все его делители. Число вводится с клавиатуры.

30. Дано целое число x . Вывести на экран все числа, на которые оно не делится. Число вводится с клавиатуры.

Порядок работы:

1. Открыть программу, указанную преподавателем, после объяснения принципов работы с ней.
2. Создать файл с расширением, указанным преподавателем.
3. Ввести текст программы.
4. Сохранить программу.
5. Выполнить компиляцию.
6. Если ошибок нет, то запустить эмуляцию программы и пошагово выполнить ее.
7. Подготовить отчет о проделанной работе.

Вопросы к теоретическому материалу

1. Что называется алфавитом?
2. Что входит в пробельные символы?
3. Что называется лексемой? Что относится к лексемам?
4. Что называется идентификатором?
5. Каким должен быть правильный идентификатор?
6. Назовите типы констант.
7. Каким образом описывается комментарий в языке Си?
8. Из каких элементов как правило состоит программа на языке Си?
9. Что называется директивами препроцессора?
10. Что определяет тип данных?
11. Перечислите основные типы данных в Си.
12. Чем различаются исполняемые и неисполняемые операторы?
13. Что называется переменной?
14. В чем разница между локальными и глобальными переменными?
15. Укажите синтаксис оператора описания переменной.
16. Чем различаются префиксная и постфиксная формы записи инкремента и декремента?
17. Укажите синтаксис оператора ветвлений.
18. Укажите синтаксис оператора ветвления *switch*.
19. Для чего применяется команда *break*?
20. Что называется циклом?
21. Укажите синтаксис оператора цикла с предусловием.
22. Укажите синтаксис оператора цикла с постусловием.
23. Укажите синтаксис оператора цикла с параметром.
24. Для чего используется оператор *continue*?

ПРОЦЕСС СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

По итогам выполнения каждой лабораторной работы студент:

1. демонстрирует преподавателю правильно работающие программы;
2. демонстрирует приобретенные знания и навыки, отвечая на несколько небольших вопросов преподавателя по составленной программе, возможностям ее доработки и теме лабораторной работы в целом;
3. демонстрирует отчет по выполненной лабораторной работе.

Итоговая оценка складывается из оценок по трем указанным составляющим.

Отчет по лабораторной работе оформляется по шаблону, представленному в приложении 1. Требования к отчету представлены в приложении 2.

ПРИЛОЖЕНИЕ 1. ШАБЛОН ОТЧЕТА
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Казанский национальный исследовательский технический университет
им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №___
по дисциплине
СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Работу выполнил

Студент гр.43___

Фамилия И.О.

Принял

Преподаватель Мартынов М.Д.

Казань 2024

1. Цель работы.

2. Задание на лабораторную работу – вставляется задание на лабораторную работу (**см приложение 3**).

3. Результат выполнения работы – формируется описание хода выполнения работы (разработанных подпрограмм, классов, переменных, структур данных и т.п.) и вставляются скриншоты с результатами работы разработанных программ (скриншоты должны быть подписаны, например, Рис. 1. Начальное состояние программы и т.п.).

4. Листинг программы – вставляется код разработанной программы с комментариями.

ПРИЛОЖЕНИЕ 2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Лист документа должен иметь книжную ориентацию, поля документа должны составлять: левое – 3 см, правое – 1 см, верхнее – 1 см, нижнее 1 см.

Нумерация страниц – внизу страницы, в правом углу, особый колонтитул для первой страницы.

Междустрочный интервал – 1,5 (полуторный), отступ первой строки – 1,25.

Текст документа должен быть выполнен с использованием шрифта Times New Roman, размер – 14, выравнивание – по ширине. Заголовки выполняются тем же шрифтом, но размера 16, полужирное начертание, размещение – по центру.

Рисунки должны размещаться по центру, они нумеруются по порядку. Перед рисунком в тексте на него должна быть ссылка. Подпись рисунка должна располагаться по центру и быть выполнена шрифтом Times New Roman, размер – 12. Сначала происходит нумерация рисунка, а затем пишется его название.

ПРИЛОЖЕНИЕ 3. ВАРИАНТЫ

Номер варианта по списку	Вариант в лабораторной (1 задание)	Вариант в лабораторной (2 задание)
1	7	22
2	6	21
3	5	20
4	4	19
5	3	18
6	2	17
7	1	16
8	15	15
9	14	14
10	13	13
11	12	12
12	11	11
13	10	10
14	9	9
15	8	8
16	7	7
17	6	6
18	5	5
19	4	4
20	3	3
21	2	2
22	1	1