

ЛАБОРАТОРНАЯ РАБОТА №3

ФУНКЦИИ. УКАЗАТЕЛИ НА ФУНКЦИЮ

Цель работы:

Приобрести умения и практические навыки для работы с функциями, с указателями на функцию и передачей функций как параметров в другие функции.

Теоретическая часть:

Функцией называется последовательность описаний и операторов, которая выполняет какое-либо законченное действие и, возможно, возвращает результат в вызывающую программу. Функция обязательно должна быть определена до ее вызова.

Синтаксис определения функции:

тип имя (список параметров) {тело функции};

Имя функции – правильный идентификатор. *Тип возвращаемого значения* может быть любым, кроме массива и функции, но может быть указателем на массив и функцию. Если функция не возвращает значение, она должна иметь тип *void*. *Список параметров* определяет значения, которые передаются в функцию при ее вызове. Элементы списка параметров имеют вид: *тип имя*, перечисление разделяется запятой. Все величины, описанные внутри функции, и параметры являются локальными по отношению к функции. Их область действия ограничена самой функцией.

Определение функции должно быть создано до ее вызова. Но есть способ определить функцию после, который заключается в использовании *описания* функции. Описание функции еще называют *прототипом*. Формальное определение прототипа выглядит следующим образом:

тип имя (список типов передаваемых параметров);

То есть, описание функции является ее заголовком – без тела функции.

Для выполнения функции ее необходимо вызвать. *Вызов* функции осуществляется следующим образом:

имя (список аргументов);

При этом при вызове функции типы данных не указываются.

В описании, определении и при вызове функции типы и порядок следования параметров должны совпадать. Имена могут быть различны, так как в определении имена игнорируются.

Возврат результатов в вызывающую программу может быть реализован следующим образом:

```
return [выражение];
```

Выражение преобразуется к типу возвращаемого значения и возвращается в точку вызова, поэтому вызов функций может выполняться внутри выражения, то есть функция может рассматриваться в качестве аргумента выражения. Если тип функции `void`, то выражение нужно опустить. Если тип функции `void` и оператор `return` по смыслу находится перед закрывающей фигурной скобкой, то его можно опустить. Также операторов `return` может быть несколько, каждый из которых завершает работу программы.

Формальные параметры – параметры, указываемые в заголовке функции.

Фактические параметры – параметры, указываемые при вызове функции (аргументы).

Передача параметров в функцию в памяти выполняется через стек. *Стек* – структура данных, которая работает по принципу LIFO. Существует два способа передачи параметров:

- по значению – в стек помещается копия значения параметра. По завершению работы стек очищается, измененное значение параметра функции не передается в вызывающую программу. Фактически, параметром, соответствующим формальному параметру, передаваемому по значению, может быть выражение (переменная или константа, в частном случае);

- по адресу – в стек передается адрес аргумента, а функция по этому адресу обращается к параметру. Все изменения, которым подвергается

параметр, сохраняются после возврата управления в вызывающую программу. Фактически, параметром, соответствующим формальному параметру, передаваемому по адресу, может быть переменная или левое допустимое выражение.

При передаче параметра по адресу производятся следующие действия:

- в заголовке формальный параметр объявляется как указатель на тип аргумента;
- в теле функции для обращения к аргументу используется операция разыменования (*);
- при вызове функции используется операция взятия адреса.

Имя функции – указатель на функцию, адрес первого байта исполняемого кода функции. Содержит адрес в кодовом сегменте, по которому располагается исполняемый код функции, то есть адрес, по которому передается управление при вызове функции.

Синтаксис:

тип (*имя) (список параметров);

Работать с указателями на функции можно следующим образом:

```
int (*f)(int, int);  
int sum(int a, int b) {return a+b;}  
int sub(int a, int b) {return a-b;}
```

Тогда f можно присвоить $f = sub$; $f = sum$;, так как имя функции – адрес первого байта кода функции, то есть f становится синонимом sub или sum .

Например, определим указатель на функцию:

```
void (*message) (void);
```

Здесь определен указатель, который имеет имя `message`. Он может указывать на функции без параметров, которые возвращают тип `void` (то есть ничего не возвращают). После названия указателя идет `(void)`, что указывает, что функция не принимает параметров.

Применим этот указатель на функцию:

```
#include <stdio.h>

void hello()
{
    printf("Hello, World \n");
}

void goodbye()
{
    printf("Good Bye, World \n");
}

int main(void)
{
    void(*message) (void);
    message=hello;
    message();
    message = goodbye;
    message();
    return 0;
}
```

Также стоит обратить внимание на скобки вокруг имени, так, например, если вернуться к использованному выше определению *void (*message) (void)*, то оно не будет аналогично определению *void *message (void)*. Во втором случае определен не указатель на функцию, а прототип функции *message*, которая возвращает указатель типа *void**.

Следует учитывать, что указатель на функцию должен по типу возвращаемого значения и типу параметров соответствовать функции, иначе он не сможет соответствовать этой функции.

Передача функции в другую функцию в качестве параметра производится так: при передаче параметров в функцию в списке формальных параметров указывают тип и имя параметров. При передаче функции в качестве фактического параметра типом этого параметра будет указатель на соответствующую функцию. Если требуется передать функцию в другую функцию в качестве параметра, необходимо:

- иметь описание функции, которую предполагается передавать в другую функцию в качестве параметра;

- описать другую функцию, в которую требуется передавать первую. Для этого в список параметров другой функции необходимо включить указатель на первую;

- вызывать другую функцию из функции `main` или другой функции, передавая в нее в качестве параметра имя первой функции.

Прототип такой функции должен иметь следующий вид:

тип (*имя (список параметров 1)) (список параметров 2);

Так должна быть описана функция, в которую передается список параметров 1 и которая возвращает в *main* указатель на функцию со списком параметров 2, который возвращает в вызывающую программу значение указанного типа.

```
# include <stdio.h>
int mul (int a, int b)
{
    return a*b;
}
int sum (int a, int b)
{
    return a+b;
}
int sub (int a, int b)
{
    return a-b;
}
int (* fb (char c)) (int ,int)
{
    switch(c)
    {
        case '*':
            return mul;
            break;
```

```
        case '+':
            return sum;
            break;
        case '-':
            return sub;
            break;
    }
}

void main() {
    printf ("Результат = %d", fb('*')(8,4));
}
```

Задание 1. Функция `main` должна вызвать указанную функцию и вывести на экран полученный результат.

1. Написать функцию, которая возвращает в вызывающую программу площадь прямоугольника. Стороны прямоугольника необходимо вводить с клавиатуры. Тип параметров – вещественный.

2. Написать функцию, которая возвращает в вызывающую программу периметр треугольника. Стороны треугольника необходимо вводить с клавиатуры. Тип параметров – вещественный. Также необходимо предусмотреть проверку, возможно ли вообще построить треугольник с указанными сторонами.

3. Написать функцию, которая возвращает в вызывающую программу периметр прямоугольника. Стороны прямоугольника необходимо вводить с клавиатуры. Тип параметров – вещественный.

4. Написать функцию, которая возвращает в вызывающую программу площадь круга радиуса r . Значение радиуса необходимо вводить с клавиатуры. Тип параметров – вещественный.

5. Написать функцию, которая возвращает в вызывающую программу длину окружности радиуса r . Значение радиуса необходимо вводить с клавиатуры. Тип параметров – вещественный.

6. Написать функцию, которая возвращает в вызывающую программу объем параллелепипеда. Значения сторон параллелепипеда необходимо вводить с клавиатуры. Тип параметров – целый.

7. Написать функцию, которая возвращает в вызывающую программу количество секунд, которое содержится в t часах. Количество часов необходимо вводить с клавиатуры. Тип параметров – целый.

8. Написать функцию, которая возвращает в вызывающую программу количество часов, которое содержится в n днях. Количество дней необходимо вводить с клавиатуры. Тип параметров – целый.

9. Написать функцию, которая вычисляет факториал переданного ей целого числа. Параметр – целое число, вводится с клавиатуры. Тип параметров – целый.

10. Написать функцию, которая определяет, является ли переданное ей целое число простым. Функция возвращает 1, если число простое, и 0 – в противном случае. Тип параметров – целый.

11. Написать функцию, которая возвращает сумму всех цифр переданного ей целого числа. Параметр – целое число, вводится с клавиатуры.

12. Написать функцию, которая возвращает корень квадратный из переданного ей вещественного числа. Использовать встроенную функцию `sqrt()` из библиотеки `math.h`.

13. Написать функцию, которая вычисляет значение n -го числа Фибоначчи. Последовательность Фибоначчи — это ряд чисел, в котором каждое следующее число равно сумме двух предыдущих. В классическом варианте последовательность начинается с 0 и 1. Параметр – целое число (порядковый номер), вводится с клавиатуры.

14. Написать функцию, которая возвращает значение n -й степени числа x . Параметры – вещественное число x и целое число n , вводятся с клавиатуры.

15. Написать функцию, которая переводит градусы по Цельсию в градусы по Фаренгейту. Параметр – вещественное число (градусы Цельсия), вводится с клавиатуры.

Задание 2. Функция `main` должна вызвать указанную функцию и вывести на экран полученный результат. Параметры передавать в функцию по адресу. Результаты возвращать через параметры, передаваемые по адресу.

1. Написать функцию, которая возвращает в вызывающую программу сумму и произведение двух целых чисел. Числа необходимо вводить с клавиатуры.

2. Написать функцию, которая возвращает в вызывающую программу квадрат и куб аргумента. Значение аргумента необходимо вводить с клавиатуры.

3. Написать функцию, которая возвращает в вызывающую программу сумму и среднее арифметическое двух целых чисел. Числа необходимо вводить с клавиатуры.

4. Написать функцию, которая возвращает в вызывающую программу наибольшее и наименьшее из трех целых чисел. Числа необходимо вводить с клавиатуры.

5. Написать функцию, которая возвращает в вызывающую программу площадь поверхности и объем куба. Значение стороны куба необходимо вводить с клавиатуры.

6. Написать функцию, которая возвращает в вызывающую программу два целых числа, между которыми находится заданное вещественное число. Число необходимо вводить с клавиатуры.

7. Написать функцию, которая вычисляет частное и остаток от деления двух целых чисел. Проверить деление на ноль перед выполнением вычислений. Число необходимо вводить с клавиатуры.

8. Написать функцию, которая вычисляет синус и косинус переданного угла (в градусах). Угол вводится с клавиатуры. Использовать функции `sin()` и `cos()` из `math.h`.

9. Написать функцию, которая вычисляет длину гипотенузы и площадь прямоугольного треугольника. Длины катетов вводятся с клавиатуры.

10. Написать функцию, которая преобразует градусы по Цельсию в

Фаренгейты и Кельвины. Значение температуры – вещественное число, вводится с клавиатуры.

11. Написать функцию, которая находит наибольший общий делитель (НОД) и наименьшее общее кратное (НОК) двух чисел. Числа вводятся с клавиатуры.

12. Написать функцию, которая возвращает корни квадратного уравнения (если они существуют). Коэффициенты уравнения вводятся с клавиатуры.

13. Написать функцию, которая генерирует n случайных чисел в заданном диапазоне (min-max) и возвращает их сумму и среднее значение через параметры.

14. Написать функцию, которая возвращает значение логарифма и экспоненты для заданного числа. Число вводится с клавиатуры. Использовать функции `log()` и `exp()` из `math.h`.

15. Написать функцию, которая возвращает количество и сумму цифр в заданном числе. Число вводится с клавиатуры.

Задание 3. Написать программу, которая реализует меню из 4 функций, которые выполняют следующие действия: возведение числа a в степень b , произведение a и b , получение остатка от деления a на b , проверка кратности чисел a и b . Все 4 функции должны принимать и возвращать целочисленные значения. Вызов нужной функции реализовать через указатель на функцию.

Задание 4. Написать функцию, которая возвращает значение выражения.

1. $f(x)=2*x;$

2. $f(x)=5+12*x;$

3. $f(x)=7-x;$

4. $f(x)=5*x-32;$

5. $f(x)=7*(1-x);$

6. $f(x)=15-4*x;$

7. $f(x)=(5-x)/x;$

8. $f(x)=x^2-1;$

9. $f(x)=x^3+x^2-1;$

10. $f(x)=x^2-2x-1.$

где x — целочисленная переменная.

Задание 5. Написать функцию *Function*, которая возвращает в вызывающую программу значение выражения в соответствии с вариантом. Здесь $f1(a)$ и $f2(a)$ – функции с одним параметром целого типа, которые возвращают целочисленное значение (используйте функции из задания 4). Передавать эти функции в *Function* при помощи указателя на функцию. Функция *main* выводит результат функции *Function*.

Варианты:

1. $f1(a)+f2(a)$;
2. $f1(a)-f2(a)$;
3. $f1(a)*f2(a)$;
4. $f1(a)/f2(a)$;
5. $(f1(a)+f2(a))/f1(a)$;
6. $1/(f1(a)+f2(a))$;
7. $(f1(a)*f2(a))-f1(a)$;
8. $1/(f1(a)*f2(a))$;
9. $f1(a)^2-f2(a)$;
10. $f1(a)^3-f2(a)^2$.

Вопросы к теоретическому материалу

1. Что называется функцией?
2. Укажите синтаксис определения функции.
3. Укажите синтаксис описания функции.
4. Укажите синтаксис вызова функции.
5. При помощи какой команды можно организовать возврат результатов в вызывающую программу?
6. Что называется формальными параметрами функции?
7. Что называется фактическими параметрами функции?
8. Какие два способа передачи параметров в функции существует?
9. Охарактеризуйте передачу параметров в функцию по значению.
10. Охарактеризуйте передачу параметров в функцию по адресу.
11. Перечислите действия, которые необходимо совершить при передаче параметра по адресу
12. Чем является имя функции?
13. Укажите синтаксис указателя на функцию.
14. Для чего можно использовать указатель на функцию?
15. Опишите процесс передачи функции в другую функцию в качестве параметра.

ПРОЦЕСС СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

По итогам выполнения каждой лабораторной работы студент:

1. Демонстрирует преподавателю правильно работающие программы;
2. Демонстрирует приобретенные теоретические знания, отвечая на пять вопросов по лабораторной работе;
3. Демонстрирует отчет по выполненной лабораторной работе, соответствующий всем требованиям.

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 Григорьева В.В.

Казань, 2025 г.

1. **Цель работы.**

2. **Задание на лабораторную работу** – вставляется задание на лабораторную работу, соответствующее индивидуальному, выданному преподавателем, варианту студента.

3. **Результат выполнения работы** – формируется описание хода выполнения работы и вставляются скриншоты с результатами работы разработанных программ (скриншоты должны быть подписаны, например, *Рисунок 1. Начальное состояние программы* и т.п.).

4. **Листинг программы** – вставляется код разработанной программы

ПРИЛОЖЕНИЕ 2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Лист документа должен иметь книжную ориентацию, поля документа должны составлять: левое – 3 см, правое – 1,5 см, верхнее – 2 см, нижнее 2 см.

Нумерация страниц – внизу страницы по центру, первая страница не нумеруется

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.