

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №6

по дисциплине: Основы программирования
тема: «Введение в функции»

Выполнил: ст. группы ПВ-223
Пахомов Владислав Андреевич

Проверили:
Притчин Иван Сергеевич
Черников Сергей Викторович

Код-ревьюер: ст. группы ПВ-223
Голуцкий Георгий Юрьевич

Белгород 2022 г.

Лабораторная работа № 6

Содержание отчёта:

- Тема лабораторной работы
- Цель лабораторной работы
- Решения задач
 - Условие задачи
 - Тестовые данные
 - Исходный код функции и её спецификация
- Работа над ошибками (код-ревью)
- Вывод по работе.

Тема лабораторной работы: Введение в функции

Цель лабораторной работы: получение навыков написания функций при решении простых задач. Закрепление навыков разработки алгоритмов разветвляющейся и циклической структуры. Получение навыков формулирования спецификаций к разрабатываемым функциям.

В некоторых функциях добавил вспомогательные файлы, их содержание выглядит следующим образом

utils.c

```
#include <math.h>

#define EPS 0.0000001

int fcompare(double a, double b) {
    return fabs(a - b) < EPS;
}
```

Заголовок: `int fcompare(double a, double b)`

Назначение: возвращает “истина” если $a = b$ с определённой точностью ($EPS = 0.0000001$), иначе - “ложь”.

Решения задач:

1. Задача №1

Напишите функцию *abs* для вычисления модуля вещественного числа *x*

Входные данные	Выходные данные
1 0	0
2 1234.6745	1234.6745
3 -12.45	12.45

1. Заголовок: `double abs(double a)`
2. Назначение: возвращает модуль значения *a*

```
#include "1func.h"

double abs(double a) {
    return a < 0 ? -a : a;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/1func.c>

2. Задача №2

Напишите функцию *sign*:

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

Входные данные	Выходные данные
1 -123123.0123	-1
2 0	0
3 42.12	1
4 0.00000000001	1
5 -0.00000000001	-1

1. Заголовок: `int sign(double x)`
2. Назначение: возвращает -1, 0 или 1 в зависимости от знака значения *x*

```
#include "2func.h"

int sign(double x) {
    return (x > 0) - (x < 0);
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/2func.c>

3. Задача №3

Напишите функцию *max2*, которая возвращает максимальное значение из двух целочисленных переменных типа *int*.

	Входные данные	Выходные данные
1	42 9	42
2	42 0	42
3	42 -12	42
4	42 42	42
5	0 123	123
6	0 0	0
7	0 -121	0
8	-140 130	130
9	-140 0	0
10	-140 -12	-12
11	-140 -1209	-140

1. Заголовок: `int max2(int a, int b)`
2. Назначение: возвращает максимальное из двух значений `a` и `b`

```
#include "3func.h"

int max2(int a, int b) {
    return a > b ? a : b;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/3func.c>

4. Задача №4

Напишите функцию *max3*, которая возвращает максимальное значение из трёх целочисленных переменных типа *int*.

Входные данные	Выходные данные
1 12 23 42	42
2 0 -124 122	122
3 0 -124 122	122
4 -12 -124 -122	-12

1. Заголовок: `int max3(int a, int b, int c)`
2. Назначение: возвращает максимальное из трёх значений `a`, `b` и `c`

```
#include "3func.h"
#include "4func.h"

int max3(int a, int b, int c) {
    return max2(a, max2(b, c));
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/4func.c>

5. Задача №5

Напишите функцию *getDistance*, которая вычисляет расстояние между двумя точками, заданными целочисленными координатами (*x1*, *y1*), (*x2*, *y2*).

Входные данные	Выходные данные
1 0 0 5 0	5.0
2 0 0 3 4	5.0
3 -11 3 -1212 90	1204.1470009
4 0 0 0 0	0.0
5 6 6 6 6	0.0

1. Заголовок: `double getDistance(int x1, int y1, int x2, int y2)`
2. Назначение: возвращает расстояние между двумя точками `A(x1; y1)` и `B(x2; y2)`

```
#include <math.h>
#include "5func.h"

double getDistance(int x1, int y1, int x2, int y2) {
    double deltaX = x1 - x2;
    double deltaY = y1 - y2;

    return sqrt(deltaX * deltaX + deltaY * deltaY);
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/5func.c>

6. Задача №6

Напишите функцию *solveX2*, которая выводит корни квадратного уравнения:

$$ax^2 + bx + c = 0 \quad (a \neq 0)$$

Входные данные	Выходные данные
1 a = -12.9 b = -100.123 c = -1000	Корней нет
2 a = -4 b = -3 c = -0.5625	-0.38
3 a = -1 b = 9 c = -8	1.00 8.00
4 a = 8.123 b = 12.1232 c = 4.6	Корней нет
5 a = 1 b = 0 c = 0	-0.00
6 a = 1.9812 b = 4.3 c = -1.123123	0.24 -2.41

1. Заголовок: `void solveX2(double a, double b, double c)`
2. Назначение: выводит корни уравнения вида $ax^2 + bx + c = 0$.

```
#include <math.h>
#include <stdio.h>
#include "utils.h"
#include "6func.h"

void solveX2(double a, double b, double c) {
    double D = b * b - 4 * a * c;

    if (fcompare(D, 0)) {
        double x = -b / (2 * a);

        printf("%.2lf\n", x);
    } else if (D > 0) {
        double sqrtD = sqrt(D);
        double x1 = (-b + sqrtD) / (2 * a);
        double x2 = (-b - sqrtD) / (2 * a);

        printf("%.2lf %.2lf\n", x1, x2);
    } else
        printf("Корней нет\n");
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/6func.c>

7. Задача №7

Написать функцию *isDigit*, которая возвращает значение 'истина', если символ *x* является цифрой, 'ложь' - в противном случае.

Входные данные	Выходные данные
1 a	false
2 0	true
3 4	true
4 9	true
5 ;	false

1. Заголовок: `bool isDigit(int x)`
2. Назначение: возвращает "истина", если символ *x* является цифрой, иначе – "ложь"

```
#include <stdbool.h>
#include "7func.h"

#define DIGIT_LOWER_BOUND '0'
#define DIGIT_UPPER_BOUND '9'

bool isDigit(char x) {
    return x >= DIGIT_LOWER_BOUND && x <= DIGIT_UPPER_BOUND;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/7func.c>

8. Задача №8

Напишите функцию *swap*, которая принимает две переменные типа *float* и обменивает их значения.

	Входные данные	Выходные данные
1	a = 5, b = 10	a = 10, b = 5
2	a = 0, b = 0	a = 0, b = 0
3	a = -123.11111, b = 42.13	a = 42.13, b = -123.11111
4	a = 0.0000001, b = 0.0000002	a = 0.0000002, b = 0.0000001

1. Заголовки: `void swap(float *a, float *b)` `void intSwap(int *a, int *b)`

2. Назначение: обменивают значения по указателям `a` и `b`

```
#include "8func.h"

void swap(float *a, float *b) {
    float c = *a;
    *a = *b;
    *b = c;
}

void intSwap(int *a, int *b) {
    int c = *a;
    *a = *b;
    *b = c;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/8func.c>

9. Задача №9

Напишите функцию *sort2*, которая упорядочивает значения *a* и *b* типа *float*.

Входные данные	Выходные данные
1 a = 10, b = 5	a = 5, b = 10
2 a = 0, b = 0	a = 0, b = 0
3 a = -123.11111, b = 42.13	a = 42.13, b = -123.11111
4 a = 0.0000002, b = 0.0000001	a = 0.0000001, b = 0.0000002
5 a = 5, b = 10	a = 5, b = 10

1. Заголовок: `void sort2(float *a, float *b) void intSort2(int *a, int *b)`
2. Назначение: упорядочивает значения по адресам `a` и `b` таким образом, что значение по адресу `a` меньше значения по адресу `b`

```
#include "8func.h"
#include "9func.h"

void sort2(float *a, float *b) {
    if (*a > *b)
        swap(a, b);
}

void intSort2(int *a, int *b) {
    if (*a > *b)
        intSwap(a, b);
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/9func.c>

10. Задача №10

Напишите функцию *sort3*, которая упорядочивает значения переменных *a*, *b*, *c* типа *float* таким образом, чтобы:

$$a \leq b \leq c$$

Входные данные	Выходные данные
1 a = 12.12 b = 45.01 c = 79.9999	a = 12.12 b = 45.01 c = 79.9999
2 a = -146.9 b = 45.01 c = -799.8	a = -799.8 b = -146.9 c = 45.01
3 a = 3 b = 2 c = 1	a = 1 b = 2 c = 3

1. Заголовки: `void sort3(float *a, float *b, float *c) void intSort3(int *a, int *b, int *c)`
2. Назначение: упорядочивает значения по адресам `a`, `b` и `c` таким образом, что значение по адресу `a` меньше значения по адресу `b`; значение по адресу `b` меньше значения по адресу `c`

```
#include "9func.h"
#include "10func.h"

void sort3(float *a, float *b, float *c) {
    sort2(a, b);
    sort2(b, c);
    sort2(a, b);
}

void intSort3(int *a, int *b, int *c) {
    intSort2(a, b);
    intSort2(b, c);
    intSort2(a, b);
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/10func.c>

11. Задача №11

Написать функцию, которая возвращает значение 'истина', если можно составить треугольник с целочисленными сторонами a, b, c ($a, b, c \in N$), 'ложь' – в противном случае

Входные данные	Выходные данные
1 $a = 3$ $b = 4$ $c = 5$	true
2 $a = 12$ $b = 1$ $c = 1$	false
3 $a = 2$ $b = 7$ $c = 1$	false
4 $a = 3$ $b = 3$ $c = 19$	false
5 $a = 3$ $b = 3$ $c = 6$	false

1. Заголовок: `bool isTrianglePossible(int a, int b, int c)`
2. Назначение: возвращает "истина", если треугольник a, b, c существует, иначе – "ложь"

```
#include <stdbool.h>
#include "10func.h"
#include "11func.h"

bool isTrianglePossible(int a, int b, int c) {
    intSort3(&a, &b, &c);

    return c < a + b;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/11func.c>

12. Задача №12

Напишите функцию *getTriangleTypeLength*, которая возвращает значение 0, если треугольник со сторонами *a*, *b*, *c* является остроугольным, 1 – если прямоугольным, 2 – тупоугольным, -1 – если треугольник с такими сторонами не существует.

Входные данные	Выходные данные
1 a = 1 b = 1 c = 13123	-1
2 a = 4 b = 4 c = 4	0
3 a = 3 b = 4 c = 5	1
4 a = 21 b = 11 c = 11	2

1. Заголовок: `int getTriangleTypeLength(int a, int b, int c)`
2. Назначение: возвращает:
 - a. -1, если треугольник со сторонами *a*, *b*, *c* не существует
 - b. 0, если треугольник со сторонами *a*, *b*, *c* остроугольный
 - c. 1, если треугольник со сторонами *a*, *b*, *c* прямоугольный
 - d. 2, если треугольник со сторонами *a*, *b*, *c* тупоугольный

```
#include "utils.h"
#include "10func.h"
#include "11func.h"
#include "12func.h"

int getTriangleTypeLength(int a, int b, int c) {
    if (isTrianglePossible(a, b, c)) {
        intSort3(&a, &b, &c);

        int bigSideSquare = c * c;
        int sumOfSquaresOfSmallerSides = a * a + b * b;

        if(fcompare(bigSideSquare, sumOfSquaresOfSmallerSides))
            return 1;
        else if (bigSideSquare > sumOfSquaresOfSmallerSides)
            return 2;
        else
            return 0;
    } else
        return -1;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/12func.c>

13. Задача №13

Напишите функцию *isPrime*, которая возвращает значение 'истина', если число является простым, иначе – 'ложь'.

Входные данные	Выходные данные
1 1	false
2 2	true
3 3	true
4 4	false
5 5	true
6 24	false
7 157	true
8 24649	false

1. Заголовок: `bool isPrime(int n)`
2. Назначение: возвращает “истина” если `n` - простое число, иначе – “ложь”.

```
#include <stdbool.h>
#include "13func.h"

bool isPrime(int n) {
    if (n == 1)
        return false;

    for (int i = 2; i < n; i++)
        if (n % i == 0)
            return false;

    return true;
}
```

```
#include <stdbool.h>
#include "13func.h"

bool isPrime(int n) {
    if (n == 1)
        return false;

    for (int i = 2; i * i <= n; i++)
        if (n % i == 0)
            return false;

    return true;
}
```

```
#include <stdbool.h>
#include "13func.h"

bool isPrime(int n) {
    if (n % 2 == 0 || n == 1)
        return n == 2 || false;

    for (int i = 3; i * i <= n; i += 2)
        if (n % i == 0)
            return false;

    return true;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/13func.c>

14. Задача №14

** Натуральное число называется совершенным, если оно равно сумме всех своих делителей, за исключением самого себя. Число 6 – совершенное, т.к. $6=1+2+3$. Число 8 – не совершенное, т.к. $8 \neq 1 + 2 + 4$. Дано натуральное число n . Получить все совершенные числа, меньшие n .

Входные данные	Выходные данные
1 1	true
2 28	true
3 27	false
4 12412	false
5 33550336	true

1. Заголовок: `bool isNumberPerfect(int n)`
2. Назначение: возвращает “истина” если n - совершенное число, иначе – “ложь”.

Входные данные	Выходные данные
1 7	1 6
2 9000	1 6 28 496 8128

1. Заголовок: `void printPerfectNumbers(int n)`
2. Назначение: выводит все совершенные числа от 1 до n включительно

```
#include <stdbool.h>
#include <stdio.h>
#include "14func.h"

bool isNumberPerfect(int n) {
    int dividersSum = 1;

    for (int i = 2; i <= n / 2; i++) {
        if (n % i == 0)
            dividersSum += i;
        if (dividersSum > n) return false;
    }

    return n == dividersSum;
}

void printPerfectNumbers(int n) {
    for (int i = 1; i <= n; i++)
        if (isNumberPerfect(i))
            printf("%d\n", i);

    printf("\n");
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/14func.c>

15. Задача №15

**** Найти количество чисел-палиндромов от 1 до n .**

Входные данные	Выходные данные
1 123	321
2 6	6
3 1234	4321

1. Заголовок: `int reverseNum(int n)`
2. Назначение: возвращает число n в обратном порядке

Входные данные	Выходные данные
1 5	true
2 121	true
3 0	true
4 1234321	true
5 9898989	true
6 476674	true
7 123	false
8 909099099	false

1. Заголовок: `bool isNumberPalindrome(int n)`
2. Назначение: возвращает “истина” если n – палиндром, иначе – “ложь”.

Входные данные	Выходные данные
1 10	9
2 100	18
3 1000	108
4 10000	198
5 100000	1098

1. Заголовок: `int countPalindromeNumbers(int n)`
2. Назначение: возвращает количество чисел-палиндромов от 1 до n включительно


```
#include <stdbool.h>
#include "15func.h"

int reverseNum(int n) {
    int reversedNum = 0;

    while (n > 0) {
        reversedNum = 10 * reversedNum + n % 10;
        n /= 10;
    }

    return reversedNum;
}

bool isNumberPalindrome(int n) {
    return n == reverseNum(n);
}

int countPalindromeNumbers(int n) {
    int palindromeCounter = 0;

    for (int i = 1; i <= n; i++)
        if (isNumberPalindrome(i))
            palindromeCounter++;

    return palindromeCounter;
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/15func.c>

16. Задача №16

**** В шестизначных автобусных билетах найти счастливые**

Входные данные	Выходные данные
1 0	0
2 111111	7
3 9023	14
4 1	1

1. Заголовок: `int sumDigits(int n)`
2. Назначение: подсчитывает сумму цифр значения `n`

Входные данные	Выходные данные
1 123123	true
2 906519	true
3 987996	true
4 100999	false

1. Заголовок: `bool isTicketHappy(int ticket)`
2. Назначение: возвращает “истина” если номер билета автобуса `ticket` счастливый, иначе – “ложь”.

```
#include <stdbool.h>
#include "16func.h"

int sumDigits(int n) {
    int sum = 0;

    while (n > 0) {
        sum += n % 10;
        n /= 10;
    }

    return sum;
}

bool isTicketHappy(int ticket) {
    int leftPair = ticket / 1000;
    int rightPair = ticket % 1000;

    return sumDigits(leftPair) == sumDigits(rightPair);
}
```

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab6/sharedfuncs/src/16func.c>

Код-ревью:

- 1) consoleoutput.h не используется
- 2) Спецификация getTriangleTypeLength не соответствует коду функции
- 3) Поменять номер лабораторной работы

Работа над ошибками (код-ревью)

- 1) Удалил
- 2) Исправил
- 3) Исправил

Вывод: в ходе выполнения лабораторной работы получены навыки написания функций при решении простых задач. Закреплены навыки разработки алгоритмов разветвляющейся и циклической структуры. Получены навыков формулирования спецификаций к разрабатываемым функциям.