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: студент группы ПВ-223
Мелехов Артём Дмитриевич

Проверили:
ст. преп. Притчин Иван Сергеевич
асс. Черников Сергей Викторович
ст. Дмитриев Андрей Александрович
ст. Сукач Руслан

Белгород 2022 г.

Лабораторная работа №6 «Введение в функции»

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

Содержание отчета:

Тема лабораторной работы

Цель лабораторной работы

Решения задач. Для каждой задачи указаны:

– Название задачи.

– Условие задачи.

– Тестовые данные.

– Исходный код функции и её спецификацию.

Вывод.

Задача №1.

Условие:

Напишите функцию abs для вычисления модуля вещественного числа x.

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1	1	Ввод положительного числа.
0	0	Ввод 0.
-5.1	5.1	Ввод отрицательного числа.

Спецификация функции abs:

1. Заголовок: `long double abs(long double a)`.
2. Назначение: возвращает модуль числа `a`.

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

Задача №2.

Условие:

Напишите функцию *sign*:

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

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
100	1	Ввод положительного числа.
0	0	Ввод 0.
-100	-1	Ввод отрицательного числа.

Спецификация функции `sign`:

1. Заголовок: `int sign(long long a)`.
2. Назначение: возвращает знак числа `a`.

```c
#include <math.h>

int sign(long long a)
{
    return a ? a / abs(a) : 0;
}
```

Задача №3.

Условие:

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

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
-1 2	2	Второй аргумент является максимальным.
3 3	3	Аргументы равны.
32 0	32	Первый аргумент является максимальным.

Спецификация функции $max2$:

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

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

Задача №4.

Условие:

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

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1 -2 3	3	Максимум не сравнивается в функции max2
1 3 2	3	Максимум сравнивается в функции max2.
3 3 1	3	В max2 попадают 2 максимума.
2 1 2	2	В данных присутствуют 2 максимума, один из которых сравнивается через max2.
2 2 2	2	Все числа равны.

Спецификация функции *max*3:

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

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

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

Задача №5.

Условие:

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

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
5 2 2 -2	5	Ввод «Пифагоровой тройки».
1 1 2 2	1.414214 (≈√2)	Ввод псевдослучайных чисел с нецелым ответом.
0 0 0 0	0	Точки *A* и *B* находятся в одном месте на плоскости.

Спецификация функции *getDistance*:

1. Заголовок: `double getDistance (int x1, int y1, int x2, int y2)`.
2. Назначение: возвращает расстояние от точки `A` с координатой (`x1`, `y1`) до точки `B` (`x2`, `y2`).

```
#include <math.h>

double getDistance (int x1, int y1, int x2, int y2)
{
    return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
}
```

Задача №6.

Условие:

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

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

Найденные корни должны быть выведены в теле функции. Если действительных корней нет, выведите соответствующее сообщение.

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1 -5 9	The quadratic equation has no roots	Дискриминант меньше 0.
1 -4 4	The root of the equation: 2	Дискриминант равен 0.
1 3 -4	The first root: 1, The second root: -4	Дискриминант больше 0.
1 -3 -4	The first root: -1, The second root: 4	Дискриминант больше 0.

Спецификация функции *solveX2*:

1. Заголовок: `void solveX2 (long long a, long long b, long long c)`.
2. Назначение: выводит корни квадратного уравнения при коэффициентах `a`, `b` и `c` или уведомления об их отсутствии, неквадратности уравнения.

```
#include <math.h>


void solveX2(long long a, long long b, long long c)
{
    long long d = pow(b, 2) - 4 * a * c;
    long long x1, x2;

    if (d >= 0)
    {
        x1 = (-b + sqrt(d)) / (2 * a);
        x2 = (-b - sqrt(d)) / (2 * a);
    }
    else
    {
        printf("The quadratic equation has no roots");

        return;
    }

    if (x1 == x2)
        printf("The root of the equation: %lld", x1);
    else
        printf("The first root: %lld,\nThe second root: %lld",
                                                    x1, x2);
}
```

Задача №7.

Условие:

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

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1	true	Ввод цифры.
j	false	Ввод не цифры.

Спецификация функции *isDigit*:

1. Заголовок: `int isDigit (int x)`.
2. Назначение: возвращает **‘истину’**, если `x` является числом, иначе **‘ложь’**.

```
#include <stdbool.h>

#define CODE_0 48

bool isDigit(char x)
{
    int code = x - CODE_0;

    return 0 <= code && code <= 9;
}
```

Задача №8.

Условие:

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

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1 2	2 1	Ввод двух чисел.

Спецификация функции *swap*:

1. Заголовок: `void swap (float *a, float *b)`.
2. Назначение: обменивает числа `a` и `b` значениями..

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

Задача №9.

Условие:

Напишите функцию $sort2$, которая упорядочивает значения a и b типа $float$. Т.е. если $a > b$ то после выполнения функции значение переменной a должно быть меньше значения переменной b.

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1 0	0 1	Первый аргумент больше второго
-10 2	-10 2	Второй аргумент больше первого.
3 3	3 3	Аргументы равны.

Спецификация функции $sort2$:

1. Заголовок: `void sort2(float *a, float *b)`.
2. Назначение: сортирует 2 числа `a` и `b` по возрастанию.

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

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

Задача №10.

Условие:

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

$a \leq b \leq c$

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1 2 3	1 2 3	Максимум является последним числом, а минимум первым.
1 3 2	1 2 3	Максимум является вторым числом, а минимум первым.
3 3 1	1 3 3	Присутствуют 2 максимума, а минимум стоит последним.
2 1 2	1 2 2	Минимум стоит вторым в последовательности. Также присутствует 2 максимума..
2 2 2	2 2 2	Все числа равны.

Спецификация функции *sort*3:

1. Заголовок: `void sort3(float *a, float *b, float *c)`.
2. Назначение: сортирует 3 числа `a`, `b` и `c` по возрастанию.

```
void swap(float* a, float* b)

{
    float t = *a;
    *a = *b;
    *b = t;
}

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

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

Задача №11.

Условие:

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.

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1 1 4	false	Отрезки из которых невозможно составить треугольник.
3 4 5	true	«Пифагорова тройка».

Спецификация функции `isTriangle`:

1. Заголовок: `bool isTriangle(int a, int b, int c)`.
2. Назначение: возвращает `‘истину’`, если из сторон `a`, `b`, и `c` можно составить треугольник, иначе `‘ложь’`.

```c
#include <stdbool.h>


void swap (int *a, int *b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

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

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

bool isTriangle(int a, int b, int c)
{
    sort3(&a, &b, &c);

    return (a + b) > c;
}
```

Задача №12.

Условие:

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.

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1 1 4	-1	Из отрезков нельзя составить треугольник.
5 4 4	0	Пример остроугольного треугольника.
3 4 5	1	Пример прямоугольного треугольника.
9 5 6	2	Пример тупоугольного треугольника.

Спецификация функции $getTriangleTypeLength$:

1. Заголовок: `int getTriangleTypeLength(int a, int b, int c)`.
2. Назначение: возвращает номер, который указывает на тип треугольника используя стороны длиной `a`, `b` и `c`.

```c
#include <math.h>

#define ACUTE_ANGLED_TRIANGLE 0
#define RIGHT_TRIANGLE 1
#define OBTUSE_TRIANGLE 2
#define NOT_A_TRIANGLE -1

void swap(int* a, int* b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

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

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

int getTriangleTypeLength(int a, int b, int c)
{
    sort3(&a, &b, &c);

    long long a2 = pow(a, 2);
    long long b2 = pow(b, 2);
    long long c2 = pow(c, 2);
    long long sumOfTheSquaresOfTheLegs = a2 + b2;

    if ((a + b) < c)
        return NOT_A_TRIANGLE;

    if (c2 < sumOfTheSquaresOfTheLegs)
        return ACUTE_ANGLED_TRIANGLE;
    else if (c2 == sumOfTheSquaresOfTheLegs)
        return RIGHT_TRIANGLE;
    else
        return OBTUSE_TRIANGLE;
}
```

Задача №13.

Условие:

Напишите функцию $isPrime$, которая возвращает значение ’истина’, если число является простым, иначе – ’ложь’. Приложите 3 вариации:

(*a*) Без оптимизаций

(*b*) С оптимизацией перебора до $\sqrt{N}$.

(*c*) С оптимизацией перебора до $\sqrt{N}$ и шагом 2.

Тестовые данные:

Входные данные	Ожидаемый результат	Пояснение
1	false	1 не является простым числом
2	true	Простое число
3	true	Простое число
4	false	Составное число
5	true	Простое число
3570	false	Составное 4-ёхзначное число.
3571	true	Простое 4-ёхзначное число.
99970	false	Составное 5-изначное число (долгое выполнение функцией *a*).
99971	true	Простое 5-изначное число (долгое выполнение функцией *a*).
3424506	false	Составное 7-изначное число (долгое выполнение функциями *a* и *b*).
3424507	true	Простое 7-изначное число (долгое выполнение функциями *a* и *b*).

Спецификация функции *isPrime*:

1. Заголовок: `bool isPrime(long long a)`.
2. Назначение: возвращает **‘истину’**, если число ***a*** простое, или **‘ложь’** в обратном случае.

(*a*)

```
#include <stdbool.h>


bool isPrime(long long a)
{
    long long d = 2;

    while (d < a && a % d)
        d++;

    return d == a;
}
```

(*b*)

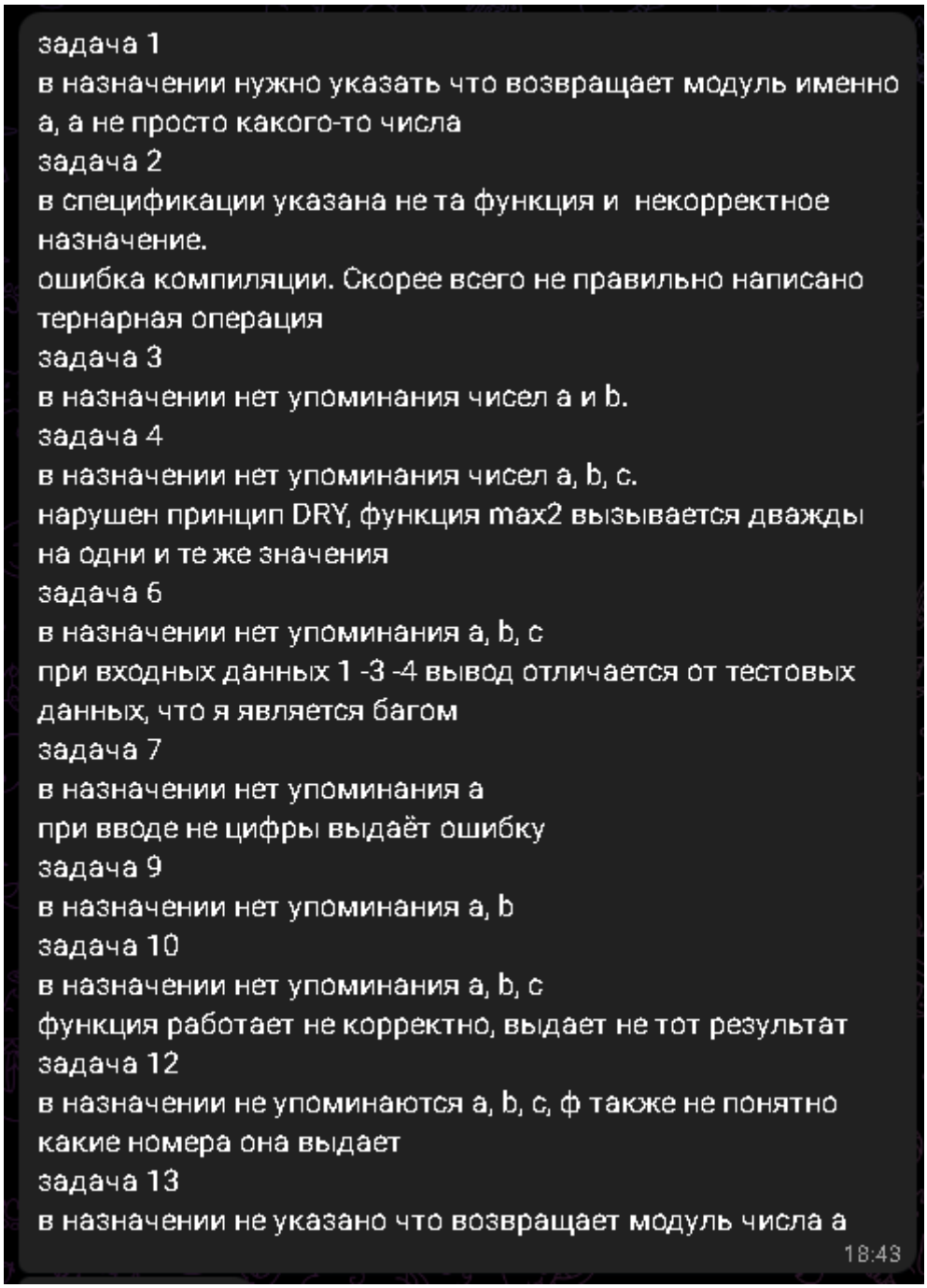
```
#include <math.h>
#include <stdbool.h>

bool isPrime(long long a)
{
    long long d = 2;
    long long maxD = (long long) sqrt(a);

    while (d <= maxD && a % d)
        d++;

    return d == (maxD + 1) && a != 1;
}
```

(*c*)

```c
#include <math.h>
#include <stdbool.h>

bool isPrime(long long a)
{
    long long d = 3;
    long long maxPotentialDivider = (long long) sqrt(a);
    long long isPrimeNumber = !(a == 1 || !(a % 2) &&
                                          a != 2);

    while (d <= maxPotentialDivider && isPrimeNumber)
    {
        isPrimeNumber = a % d;
        d += 2;
    }

    return isPrimeNumber;
}
```

Вывод: в ходе работы получены навыки написания функций.

Ревью:


задача 1
в назначении нужно указать что возвращает модуль именно a, а не просто какого-то числа
задача 2
в спецификации указана не та функция и некорректное назначение.
ошибка компиляции. Скорее всего не правильно написано тернарная операция
задача 3
в назначении нет упоминания чисел a и b.
задача 4
в назначении нет упоминания чисел a, b, c.
нарушен принцип DRY, функция max2 вызывается дважды на одни и те же значения
задача 6
в назначении нет упоминания a, b, c
при входных данных 1 -3 -4 вывод отличается от тестовых данных, что я является багом
задача 7
в назначении нет упоминания a
при вводе не цифры выдаёт ошибку
задача 9
в назначении нет упоминания a, b
задача 10
в назначении нет упоминания a, b, c
функция работает не корректно, выдает не тот результат
задача 12
в назначении не упоминаются a, b, c, ф также не понятно какие номера она выдает
задача 13
в назначении не указано что возвращает модуль числа a
18:43

• в пособии сказано, что в назначение спецификации переменные должны быть в том же стиле что и код, можешь воспользоваться форматом по образцу, можешь копировать из ф-ии
• во многих задачах можно добавить проверки с отрицательным числами
• Задача 1: вещественное число – число с плавающей точкой
• Задача 2: в данном случае типы данных должны соответствовать, тк при больших числах ответ будет неверным
• Задача 3: скорее возвращает 1-у со знаком введённого числа
• Задача 4: можно написать проще, передав в max2 max2(a, b) и c
• Задача 6: заголовок в спецификации указан неверно; в условии задано, что a не равно 0, поэтому проверка не нужна
• Задача 7: маг константа; я бы проверил на крайние значения; лучше записать <переменная <= число>
• Задача 11: заголовок в спецификации указан неверно; неизвестно что обозначает A в названии ф-ии
• Задача 12: можешь сразу возвращать значения не используя переменную flag; рассчитай сумму заранее (можешь даже найти корень суммы, его можно адекватно сравнивать с целыми числами)
• Задача 13: предоставь тесты для чисел от 1-го до 5-и; также в задаче сказано, что вводится число, те оно может быть как и положительным, так и отрицательным; 1 у тебя не является простым числом, я думаю, что стоит указать это в спецификации; делитель переводится как divider, а не divinder

изменено 16:13