|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №1  по дисциплине «Языки программирования»  Вариант 1 | | |
|  | | Работу выполнил  студент группы ПМИ-9,10-2023 2 курса  Казаков Н.С.  «02» Октября 2024 г. |
| Работу проверил  Фамилия И.О. преподавателя  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

**Задание 1 5**

**Текст задания 5**

**Алгоритм решения 5**

**Тестирование 5**

**Код программы 5**

**Задание 2 6**

**Текст задания 6**

**Алгоритм решения 6**

**Тестирование 6**

**Код программы 6**

**Задание 3 7**

**Текст задания 7**

**Алгоритм решения 7**

**Тестирование 7**

**Код программы 7**

**Задание 4 8**

**Текст задания 8**

**Алгоритм решения 8**

**Тестирование 8**

**Код программы 8**

**Задание 5 9**

**Текст задания 9**

**Алгоритм решения 9**

**Тестирование 9**

**Задание 6 10**

**Текст задания 10**

**Алгоритм решения 10**

**Тестирование 10**

**Задание 7 11**

**Текст задания 11**

**Алгоритм решения 11**

**Тестирование 11**

**Код программы 11**

**Задание 8 12**

**Текст задания 12**

**Алгоритм решения 12**

**Тестирование 12**

**Код программы 12**

**Задание 9 13**

**Текст задания 13**

**Алгоритм решения 13**

**Тестирование 13**

**Задание 10 14**

**Текст задания 14**

**Алгоритм решения 14**

**Тестирование 14**

**Задание 11 15**

**Текст задания 15**

**Алгоритм решения 15**

**Тестирование 15**

**Код программы 15**

**Задание 12 16**

**Текст задания 16**

**Алгоритм решения 16**

**Тестирование 16**

**Код программы 17**

**Задание 13 17**

**Текст задания 17**

**Алгоритм решения 17**

**Тестирование 17**

**Код программы 17**

**Задание 14 18**

**Текст задания 18**

**Алгоритм решения 19**

**Тестирование 19**

**Код программы 20**

**Задание 15 20**

**Текст задания 20**

**Алгоритм решения 20**

**Тестирование 20**

**Код программы 20**

# Задание 1

## Текст задания

## Дробная часть. Дана сигнатура функции: double fraction (double x); Необходимо реализовать функцию таким образом, чтобы она возвращала только дробную часть числа х. Подсказка: вещественное число может быть преобразовано к целому путем отбрасывания дробной части.

## Алгоритм решения

1. Входные данные:

- Принимаем одно вещественное число x в качестве аргумента функции.

2. Отбрасывание целой части:

- Преобразуем x к целому числу, чтобы получить только целую часть. Это можно сделать с помощью приведения типов (например, с использованием static\_cast<int>(x)).

3. Вычисление дробной части:

 - Вычисляем дробную часть, вычитая целую часть из оригинального числа: fraction = x - static\_cast<int>(x).

## Тестирование

## 

## Код программы

[KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 2

## Текст задания

Двузначное. Дана сигнатура функции: bool is2Digits (int x); Необходимо реализовать функцию таким образом, чтобы она принимала число x и возвращала true, если оно двузначное.

## Алгоритм решения

1. Входные данные:

- Принимаем целое число x в качестве аргумента функции.

2. Обработка отрицательных чисел:

- Используем функцию abs(x) для получения абсолютного значения числа. Это позволяет нам работать с отрицательными числами, не нарушая логику проверки.

3. Проверка диапазона:

- Проверяем, находится ли абсолютное значение числа в диапазоне от 10 до 99 (включительно). Если да, то число двузначное.

## Тестирование

## 

## Код программы

[KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 3

## Текст задания

## Диапазон. Дана сигнатура функции: bool isInRange (int a, int b, int num); Функция принимает левую и правую границу (a и b) некоторого числового диапазона. Необходимо реализовать функцию таким образом, чтобы она возвращала true, если num входит в указанный диапазон (включая границы). Обратите внимание, что отношение a и b заранее неизвестно (неясно кто из них больше, а кто меньше)

## Алгоритм решения

1. Входные данные:

- Принимаем три целых числа: a и b — границы диапазона, и num — число, которое нужно проверить.

2. Определение границ:

- Используем условный оператор для определения меньшей (нижней) и большей (верхней) границы:

- Если a < b, то lower = a и upper = b.

-Если a >= b, то lower = b и upper = a.

3. Проверка принадлежности:

- Проверяем, входит ли число num в диапазон [lower, upper]. Это делается с помощью условия:

- num >= lower && num <= upper.

**Тестирование**

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание **Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

## Код программы

# [KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 4

## Текст задания

Равенство. Дана сигнатура функции: bool isEqual(int a, int b, int c); Необходимо реализовать функцию таким образом, чтобы она возвращала true, если все три полученных функцией числа равны

## Алгоритм решения

1. Входные данные:

- Принимаем три целых числа: a, b и c.

2. Проверка равенства:

- Используем логическое выражение для проверки равенства всех трех чисел:

- a == b && b == c.

## Тестирование

## Изображение выглядит как снимок экрана, линия, текст Автоматически созданное описание

## Код программы

# [KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 5

## Текст задания

Модуль числа. Дана сигнатура функции: int abs (int x); Необходимо реализовать функцию таким образом, чтобы она возвращала модуль числа х (если оно было положительным, то таким и остается, если он было отрицательным – то необходимо вернуть его без знака минус).

## Алгоритм решения

1. Входные данные:

- Принимаем одно целое число x.

2. Проверка знака:

- Если x больше или равно нулю, возвращаем x.

- Если x меньше нуля, возвращаем -x (то есть, меняем знак на противоположный).

## Тестирование

## 

## Код программы

# [KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 6

## Текст задания

Тридцать пять. Дана сигнатура функции: bool is35 (int x); Необходимо реализовать функцию таким образом, чтобы она возвращала true, если число x делится нацело на 3 или 5. При этом, если оно делится и на 3, и на 5, то вернуть надо false. Подсказка: оператор % позволяет получить остаток от деления.

## Алгоритм решения

1.Входные данные: Получаем целое число x.

2. Проверка делимости:

- Используем оператор % для проверки делимости:

- Проверьте, делится ли x на 3: x % 3 == 0.

- Проверьте, делится ли x на 5: x % 5 == 0.

3. Логика возврата:

- Если x делится и на 3, и на 5 (т.е. оба условия истинны), верните false.

- Если x делится только на 3 или только на 5 (т.е. хотя бы одно из условий истинно), верните true.

- Если x не делится ни на 3, ни на 5, верните false.

## Тестирование

## 

## Код программы

# [KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 7

## Текст задания

Тройной максимум. Дана сигнатура функции: int max3 (int x, int y, int z); Необходимо реализовать функцию таким образом, чтобы она возвращала максимальное из трех полученных функцией чисел. Подсказка: идеальное решение включает всего две инструкции if и не содержит вложенных if

## Алгоритм решения

1. Сравнение двух чисел: Сначала сравниваем два числа X и Y и находим большее из них.

2. Сравнение с третьим числом: Затем сравниваем результат предыдущего сравнения с третьим числом (z).

## Тестирование

## 

## Код программы

# [KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 8

## Текст задания

Двойная сумма. Дана сигнатура функции: int sum2 (int x, int y); Необходимо реализовать функцию таким образом, чтобы она возвращала сумму чисел x и y. Однако, если сумма попадает в диапазон от 10 до 19, то надо вернуть число 20.

## Алгоритм решения

1. Сложить два числа: Сначала вычисляем сумму x и y.

2. Проверить диапазон: Если сумма попадает в диапазон от 10 до 19 (включительно), возвращаем 20.

3. Вернуть сумму: Если сумма не попадает в указанный диапазон, возвращаем ее.

## Тестирование

## Изображение выглядит как текст, Шрифт, снимок экрана, линия Автоматически созданное описание

## Код программы

# [KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 9

## Текст задания

День недели. Дана сигнатура функции: String day (int x); Функция принимает число x, обозначающее день недели. Необходимо реализовать функцию таким образом, чтобы она возвращала строку, которая будет обозначать текущий день недели, где 1 — это понедельник, а 7 – воскресенье. Если число не от 1 до 7 то верните текст “это не день недели”. Вместо if в данной задаче используйте switch.

## Алгоритм решения

1. Определение входного параметра: Функция принимает целое число x, которое обозначает день недели.

2. Использование switch: В зависимости от значения x, функция будет возвращать соответствующую строку с названием дня недели.

3. Обработка некорректных значений: Если x не находится в диапазоне от 1 до 7, функция должна вернуть строку "это не день недели".

## Тестирование

## 

## Код программы

# [KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 10

## Текст задания

Числа подряд. Дана сигнатура функции: String listNums (int x); Необходимо реализовать функцию таким образом, чтобы она возвращала строку, в которой будут записаны все числа от 0 до x (включительно).

## Алгоритм решения

1. Определение входного параметра: Функция принимает целое число x.

2. Инициализация строки: Создайте пустую строку, в которую будут добавляться числа.

3. Цикл от 0 до x: Используйте цикл для перебора всех чисел от 0 до x.

4. Добавление чисел: Добавляйте каждое число к строке, добавляя пробел (или запятую) после каждого числа.

5. Удаление лишнего пробела: Убедитесь, что в конце строки не остается лишнего пробела или разделителя.

6. Возврат результата: Верните сформированную строку.

## Тестирование

## 

## Код программы

# [KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 11

## Текст задания

Четные числа. Дана сигнатура функции: String chet (int x); Необходимо реализовать функцию таким образом, чтобы она возвращала строку, в которой будут записаны все четные числа от 0 до x (включительно). Подсказа для обеспечения качества кода: инструкцию if использовать не следует.

## Алгоритм решения

1. Определение входного параметра: Функция принимает целое число x.

2. Инициализация строки: Создайте пустую строку для хранения результатов.

3. Цикл от 0 до x: Используйте цикл для перебора всех чисел от 0 до x.

4. Проверка четности: Вместо if используйте арифметическую операцию для добавления только четных чисел (например, шаг цикла может быть равен 2).

5. Добавление чисел: Добавляйте каждое четное число к строке.

6. Возврат результата: Верните сформированную строку.

## Тестирование

## 

## Код программы

[KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 12

## Текст задания

Длина числа. Дана сигнатура функции: int numLen (long x); Необходимо реализовать функцию таким образом, чтобы она возвращала количество знаков в числе x.

## Алгоритм решения

1. Ввод числа:

- Получите целое число от пользователя. Это может быть как положительное, так и отрицательное число.

2. Проверка на ноль:

- Если число равно 0, верните 1, так как у числа 0 есть одна цифра.

3. Преобразование в абсолютное значение:

- Если число отрицательное, преобразуйте его в положительное (например, используя функцию абсолютного значения).

4. Инициализация счетчика:

- Создайте переменную для подсчета цифр и установите ее равной 0.

5. Цикл для подсчета цифр:

- Пока число больше 0:

- Разделите число на 10 (это удалит последнюю цифру).

- Увеличьте счетчик на 1.

6. Вывод результата:

- Верните значение счетчика, которое будет равно количеству цифр в числе.

## Тестирование

## 

## Код программы

[KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 13

## Текст задания

Квадрат. Дана сигнатура функции: void square (int x); Необходимо реализовать функцию таким образом, чтобы она выводила на экран квадрат из символов ‘\*’ размером х, у которого х символов в ряд и х символов в высоту

## Алгоритм решения

1. Ввод параметра:

- Принимаем целое число x, которое определяет размер квадрата.

2. Проверка на положительность:

- Если x меньше 0 выводим сообщение об ошибке и завершаем выполнение функции.

3. Цикл для высоты:

- Используем внешний цикл, который будет повторяться x раз (для каждой строки квадрата).

4. Цикл для ширины:

- Внутри внешнего цикла используем внутренний цикл, который также будет повторяться x раз (для каждого символа в строке).

- На каждой итерации внутреннего цикла выводим символ \*.

5. Перевод строки:

- После завершения внутреннего цикла (после вывода x символов \*) выводим символ новой строки для перехода на следующую строку.

## Тестирование

## Изображение выглядит как снимок экрана, текст, Шрифт, линия Автоматически созданное описание

## Код программы

[KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 14

## Текст задания

Правый треугольник. Дана сигнатура функции: void rightTriangle (int x); Необходимо реализовать функцию таким образом, чтобы она выводила на экран треугольник из символов ‘\*’ у которого х символов в высоту, а количество символов в ряду совпадает с номером строки, при этом треугольник выровнен по правому краю. Подсказка: перед символами ‘\*’ следует выводить необходимое количество пробелов.

## Алгоритм решения

1. Ввод параметра:

- Функция принимает одно целое число x, которое представляет высоту треугольника.

2. Проверка на положительность:

- Если x меньше 0 выводим сообщение об ошибке и завершаем выполнение функции.

3. Цикл по высоте:

- Используем внешний цикл, который будет проходить от 1 до x (включительно). Каждая итерация соответствует номеру строки треугольника.

4. Вывод пробелов:

- Для каждой строки выводим необходимое количество пробелов, чтобы выровнять символы \* по правому краю. Количество пробелов будет равно x - i, где i — номер текущей строки.

5. Вывод символов \*:

- После пробелов выводим символы \*. Количество символов в текущей строке будет равно i.

6. Перевод строки:

- После завершения вывода символов для текущей строки добавляем символ новой строки для перехода на следующую строку.

## Тестирование

## Изображение выглядит как снимок экрана, текст, Шрифт, линия Автоматически созданное описание

## Код программы

[KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)

# Задание 15

## Текст задания

Поиск максимального. Дана сигнатура функции: int maxAbs (int arr[]); Необходимо реализовать функцию таким образом, чтобы она возвращала наибольшее по модулю (то есть без учета знака) значение массива arr.

## Алгоритм решения

1. Инициализировать переменную для хранения максимального по модулю значения (например, max\_value) с начальным значением 0 или с первым элементом массива, если он существует.

2. Пройтись по всем элементам массива:

- Для каждого элемента вычислить его модуль (используя функцию abs).

- Если модуль текущего элемента больше, чем текущее значение max\_value, обновить max\_value.

3. Вернуть max\_value после завершения обхода массива.

## Тестирование

## Изображение выглядит как текст, снимок экрана, Шрифт, линия Автоматически созданное описаниеИзображение выглядит как снимок экрана, текст, Шрифт, линия Автоматически созданное описание

## Код программы

[KNs4/LAB-1-C- (github.com)](https://github.com/KNs4/LAB-1-C-)