|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по индивидуальной работе №1  по дисциплине «Языки программирования»  Вариант 1 | | |
|  | | Работу выполнил  студент группы ИТ-8, 2 курса  Лысков Никита Анатольевич  «23» 09.2025 г. |
| Работу проверил/a  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фамилия И.О.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Постановка задачи 1 3](#_Toc209561350)

[Алгоритм решения 3](#_Toc209561351)

[Тестирование 3](#_Toc209561352)

[Постановка задачи 2 3](#_Toc209561353)

[Алгоритм решения 3](#_Toc209561354)

[Тестирование 4](#_Toc209561355)

[Постановка задачи 3 4](#_Toc209561356)

[Алгоритм решения 4](#_Toc209561357)

[Тестирование 4](#_Toc209561358)

[Постановка задачи 4 5](#_Toc209561359)

[Алгоритм решения 5](#_Toc209561360)

[Тестирование 5](#_Toc209561361)

[Код программы 5](#_Toc209561362)

# Постановка задачи 1

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. Получить число с плавающей запятой x.
2. Преобразовать число x к целому типу, отбросив дробную часть
3. Вычесть целую часть из исходного числа x, чтобы получить только дробную часть.
4. Вернуть результат.

# Тестирование



# Постановка задачи 2

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. Получить число x.
2. Проверить, отрицательно ли число:
   * Если x < 0, вернуть −*x*.
   * Иначе вернуть x как есть.
3. Вернуть абсолютное значение числа

# Тестирование



# Постановка задачи 3

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. Получить целое число x.
2. Если *x* ≥ 0:
   * От 0 до x включительно пройти циклом по целым числам.
   * В каждой итерации добавить число в строку с пробелом.
3. Если  *x* < 0:
   * Начать цикл от 0 и уменьшать до x (включительно).
   * В каждой итерации добавить текущее число в строку с пробелом.
4. Вернуть сформированную строку с перечисленными числами.

# Тестирование



# Постановка задачи 4

Дана сигнатура функции: int findFirst (int arr[], int x); Необходимо реализовать функцию таким образом, чтобы она возвращала индекс первого вхождения числа x в массив arr. Если число не входит в массив – возвращается -1.

# Алгоритм решения

1. Иметь фиксированный массив чисел и число для поиска x.
2. Пройти по массиву поэлементно, сравнивая каждый элемент с x.
3. Если элемент совпал с x, вернуть индекс текущей позиции.
4. Если после проверки всех элементов совпадений нет, вернуть -1.

# Тестирование



# Код программы

#include <iostream>

#include <string>

double fraction(double x) // 1

{

int y = static\_cast<int>(x);

return x - y;

}

int abss(double x) // 2

{

return (x < 0) ? -x : x;

}

std::string listNums(int x) // 3

{

std::string result{};

if (x >= 0)

{

for (int i{}; i <= x; ++i)

{

result += std::to\_string(i) + ' ';

}

return result;

}

else if (x < 0)

{

for (int i{}; i != x - 1; --i)

{

result += std::to\_string(i) + ' ';

}

return result;

}

}

int findFirst(int arr[], int x) // 4

{

int sizeArray = 7;

for (int i{}; i < sizeArray; ++i)

{

if (arr[i] == x)

return i;

}

return -1;

}

int main() {

setlocale(LC\_ALL, "rus");

int task{};

double x{};

int arr[]{ 1,2,3,4,2,2,5 };

while (true) {

std::cout << "Введите номер задания (1-4): ";

std::cin >> task;

if (std::cin.fail()) {

std::cin.clear();

std::cin.ignore(32767, '\n');

std::cout << "Ошибка ввода! Введите целое число." << std::endl;

}

else if (task < 1 || task > 4) {

std::cin.ignore(32767, '\n');

std::cout << "Данной задачи не существует! Введите число от 1 до 4." << std::endl;

}

else {

break;

}

}

while (true)

{

std::cout << "x = ";

std::cin >> x;

if (std::cin.fail())

{

std::cin.clear();

std::cin.ignore(32767, '\n');

std::cout << "Ошибка ввода! Введите число." << std::endl;

}

else

{

break;

}

}

std::cout << "Результат: ";

switch (task) {

case 1: {

std::cout << fraction(x);

}

break;

case 2:

{

std::cout << abss(x);

}

break;

case 3:

{

std::cout << listNums(static\_cast<int>(x));

}

break;

case 4:

{

std::cout << findFirst(arr, static\_cast<int>(x));

}

break;

default:

{

std::cout << "Такого задания нет" << std::endl;

}

};

return EXIT\_SUCCESS;

}