Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Кафедра инженерной психологии и эргономики

Учебная дисциплина «Основы Алгоритмизации и программирования»

ОТЧЕТ

по лабораторной работе №8

«Функции»

Вариант 23

Подготовил: Меньшиков М.А.

Проверил: Усенко Ф.В.

Минск 2024

***Цель работы:*** сформировать навыки и умения обработки структурированных типов данных, организованных в виде функций.

***Задание 1:*** Напишите программу вычитания двух обыкновенных несократимых дробей  и  Результат представить в виде несократимой дроби.

#include <iostream>

using namespace std;

int gcd(int num, int den) {

while (den != 0) {

int temp = den;

den = num % den;

num = temp;

}

return num;

}

double numerator(double m, double n, double p, double q) {

int num = m \* q - p \* n;

int den = n \* q;

int fracGCD = gcd(num, den);

int reduced\_num = num / fracGCD;

return reduced\_num;

}

double denominator(double m, double n, double p, double q) {

int den = n \* q;

int num = m \* q - p \* n;

int fracGCD = gcd(num, den);

int reduced\_den = den / fracGCD;

return reduced\_den;

}

void output(int m, int n, int p, int q) {

int reduced\_num = numerator(m, n, p, q);

int reduced\_den = denominator(m, n, p, q);

if (reduced\_den == 0) {

cout << m << '/' << n << " - " << p << '/' << q << " = " << "nothing";

}

else {

if (reduced\_num == 0) {

cout << m << '/' << n << " - " << p << '/' << q << " = " << "0";

}

else {

cout << m << '/' << n << " - " << p << '/' << q << " = " << reduced\_num << '/' << reduced\_den;

}

}

}

int main() {

double m, n, p, q;

cout << "Enter numerator and denominator of 1st fraction: " << endl;

cin >> m >> n;

cout << "Enter numerator and denominator of 2nd fraction: " << endl;

cin >> p >> q;

output(m, n, p, q);

return 0;

}

Результаты работы программы на рисунках 1-3

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1 – работа программы при вводе обычных дробей

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, программное обеспечение, текст

Автоматически созданное описание

Рисунок 2 – работа программы с результатом равным 0

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Рисунок 3 – работа программы с нулевым знаменателем

Блок-схемы работы программы представлены на рисунках 4-8

Изображение выглядит как текст, Шрифт, диаграмма, снимок экрана

Автоматически созданное описание

Рисунок 4 – функция для поиска НОДа

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 5 – функция для определения числителя

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 6 – функция для определения знаменателя

Изображение выглядит как текст, диаграмма, рукописный текст, Шрифт

Автоматически созданное описание

Рисунок 7 – функция для вывода результата

Изображение выглядит как текст, Шрифт, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 8 – функция main

***Задание 2:*** а) для последовательности, начинающейся на четное число, выполняет циклический сдвиг влево на количество элементов равное первому элементу последовательности.

б) для последовательности, начинающейся на нечетное число, выполняет циклический сдвиг вправо на количество элементов равное последнему элементу последовательности.

#include <iostream>

using namespace std;

void arrayShift(int even[], int size, int random) {

int shift = even[0];

for (int j = 0; j < shift; j++) {

int temp = even[0];

for (int i = 0; i < size - 1; i++) {

even[i] = even[i + 1];

}

even[size - 1] = temp;

}

}

void arrayShift(int odd[], int size) {

int shift = odd[size - 1];

for (int j = 0; j < shift; j++) {

int temp = odd[size - 1];

for (int i = size - 1; i > 0; i--) {

odd[i] = odd[i - 1];

}

odd[0] = temp;

}

}

int main() {

int size;

cout << "Enter the size of array: ";

cin >> size;

int\* ptrarr = new int[size];

for (int i = 0; i < size; i++) {

cin >> ptrarr[i];

}

if (ptrarr[0] % 2 == 0) {

arrayShift(ptrarr, size, 1);

}

else {

arrayShift(ptrarr, size);

}

for (int i = 0; i < size; i++) {

cout << ptrarr[i];

}

delete[] ptrarr;

return 0;

}

Результаты работы программы на рисунках 9-10

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Рисунок 9 – работа программы со сдвигом вправо

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 10 – работа программы со сдвигом влево

Блок-схемы работы программы представлены на рисунках 11-13

Изображение выглядит как текст, Шрифт, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 11 – функция для сдвига влево

Изображение выглядит как текст, Шрифт, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 12 – функция для сдвига вправо

Изображение выглядит как текст, диаграмма, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 13 – функция main

**Вывод:** В ходе лабораторной работы мы изучили применение функций для обработки структурированных данных. Мы реализовали алгоритмы для циклического сдвига последовательностей, в зависимости от четности первого элемента: сдвиг влево для четного числа и сдвиг вправо для нечетного. Это позволило нам лучше понять, как функции могут модифицировать данные и выполнять различные операции на основе условий. Кроме того, мы разработали программу для вычитания двух обыкновенных несократимых дробей, что углубило наши знания о работе с дробями и арифметическими операциями. Результат был представлен в виде несократимой дроби, что потребовало от нас реализации алгоритмов сокращения. Эти задачи помогли развить навыки программирования и понимание важности функций в организации кода. Полученные знания будут полезны для решения более сложных задач в будущем, связанных с обработкой данных и математическими вычислениями.