Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«ИЗУЧЕНИЕ ОДНОМЕРНЫХ МАССИВОВ И СТРОК»**

**ПО ДИСЦИПЛИНЕ**

**«ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-205-52-00

Ильин Тимофей Анатольевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Цель работы: получить базовые навыки работы с одномерными массивами, освоить принципы работы со строками как с частным случаем одномерных массивов.

**Задание 1 (Вариант 5)**

Написать программу на PascalABC программу, выполняющую поиск такого минимального k, что сумма первых k элементов массива превосходит заданное число s.

**Описание алгоритма задания 1**

1. Ввод переменной k (определение размера массива);
2. Ввод цикла, выполняющий действия от 1 до k;
3. Ввод переменной arr[i] (ввод элементов массива);
4. Ввод значение переменной s;
5. Присваивание к переменной sum (сумма элементов массива) значение 0 и к переменной i значение 1;
6. Ввод цикла с предусловием, который будет выполняет действия тогда и только тогда, когда условие: sum меньше или равно заданного числа s и i меньше или равно размеру массива верно;
7. Присваивание к переменной sum значение sum + arr[i] (элемент массива) и к переменной i значение i+1;
8. Если sum больше заданного числа s, то вывод i-1(предыдущий элемент массива);
9. Если sum меньше заданного числа s, то вывод 'Не удалось найти такое k'

**Схема алгоритма с комментариями задания 1**

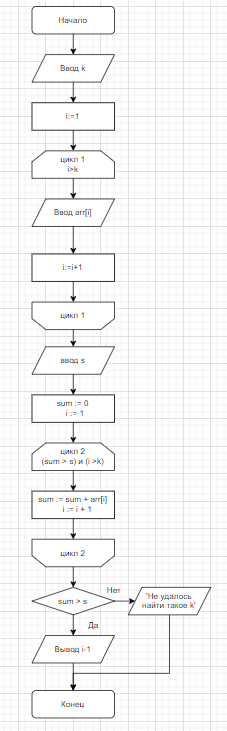


Рис.1 Схема алгоритма первой задачи

**Код программы задания 1**

**program** zad1;

**var**

arr: **array**[1..100] **of** integer;

s, i, k, sum: integer;

**begin**

write('Введите размер массива: ');

read(k);

writeln('Введите элементы массива: ');

**for** i := 1 **to** k **do**

read(arr[i]);

write('Введите заданное число s: ');

read(s);

sum := 0;

i := 1;

**while** (sum <= s) **and** (i <= k) **do**

**begin**

sum := sum + arr[i];

i := i + 1;

**end**;

**if** sum > s **then**

writeln('Минимальное k:', i - 1)

**else**

writeln('Не удалось найти такое k');

**end**.

**Результат выполнения программы задания 1**

1. Результат программы с найденным k. Размер массива вводим 5. Элементы массива: 2 3 5 6 8. Заданное число: 6

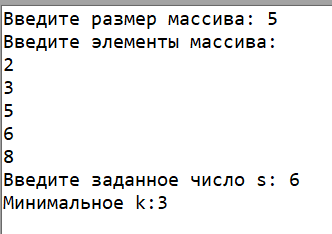


Рис. 2 Результат программы задания 1(Удалось найти такое k)

1. Результат программы где такого числа k нет. Размер массива вводим 5. Элементы массива: 2 3 5 6 8. Заданное число: 34

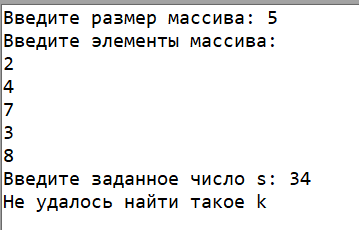


Рис. 3 Результат программы задания 1(Не удалось найти такое k)

**Задание 2 (Вариант 5)**

Написать программу, выполняющую в строке замену заданной подстроки на инверсную ей.

**Описание алгоритма задания 2**

1. Ввод переменной str;
2. Ввод переменной inv;
3. Присвоение к переменной i значения 1, для того чтобы цикл начинался с 1;
4. Ввод цикла с предусловием, который будет выполняет действия тогда и только тогда, когда условие: i <= length(str) верно.
5. Присвоение к переменным j и k значение 1;15
6. Ввод цикла с предусловием, который будет выполняет действия тогда и только тогда, когда условие: j <= length(inv)) **and** (k <= length(str)) **and** (str[k] = inv[j]) верно.
7. Присвоение к переменной j значение 1 и к переменной k значение переменной i, для работы цикла;
8. Если j > length(inv), то вводим обратный цикл от length(inv) до 1, к переменной n присваиваем значение n + inv[y] и к переменной i значение i + length(inv);
9. Если j <= length(inv), то к переменной n присваиваем значение n + str[i] и к переменной i значение i + 1;
10. Вывод переменной n.

**Схема алгоритма с комментариями задания 2**

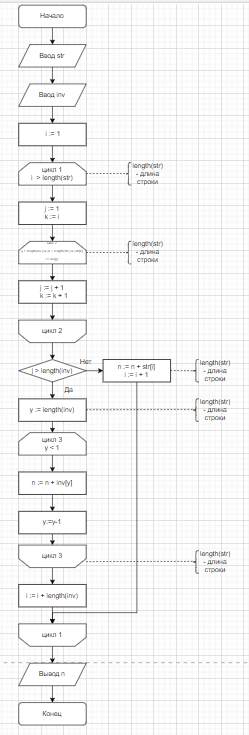
****

Рис. 4 Схема алгоритма задачи 2

**Код программы задания 2**

**program** zad2;

**var**

str, inv, n: string;

i, j, k, y: integer;

**begin**

writeln('Введите строку: ');

readln(str);

writeln('Введите подстроку для инверсии: ');

readln(inv);

i := 1;

**while** i <= length(str) **do**

**begin**

j := 1;

k := i;

**while** (j <= length(inv)) **and** (k <= length(str)) **and** (str[k] = inv[j]) **do**

**begin**

j := j + 1;

k := k + 1;

**end**;

**if** j > length(inv) **then**

**begin**

**for** y := length(inv) **downto** 1 **do**

**begin**

n := n + inv[y];

**end**;

i := i + length(inv);

**end**

**else**

**begin**

n := n + str[i];

i := i + 1;

**end**;

**end**;

writeln(n);

**end**.

**Результат выполнения программы задания 2**

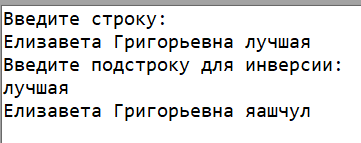


Рис. 5 Результат программы задания 2

Вывод: в результате выполнения домашней контрольной работы были получены базовые навыки работы с одномерными массивами и освоены принципы работы с строками, которые являются частным случаем одномерных массивов.

Первое задание с одномерным массивом было выполнено без проблем. Однако во втором задании с использованием строк возникли проблемы при написании кода, так как условие запрещало использовать стандартные функции для работы со строками, за исключением функции определения длины строки - length. Это усложнило задачу, однако она была выполнима.

Было необходимо реализовать различные операции со строками, такие как объединение строк, поиск подстроки, замена символов, сравнение строк и т.д. Для достижения нужной функциональности пришлось использовать циклы и условные операторы. Благодаря этому удалось успешно выполнить задание, продемонстрировав понимание основных принципов работы с одномерными массивами в контексте строк.

Выполнение домашней работы также способствовало развитию навыков аналитического мышления и умения находить решение проблемы, используя доступные инструменты. Была затронута тема оптимизации работы с массивами и строками, что позволило сделать код более эффективным и понятным.