Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №2**

**«ИЗУЧЕНИЕ ОДНОМЕРНЫХ МАССИВОВ И СТРОК»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ ФЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Гончаров Мирослав Даниилович

Преподаватель:

Кузьминых Ангелина Владимировна

Киров

2023

**Цель работы**

Цель данной лабораторной работы получить базовые навыки работы с одномерными массивами, освоить принципы работы со троками как с частным случаем одномерных массивов.

**Формулировка задания (с вариантом)**

Вариант: 1

Задание 1:

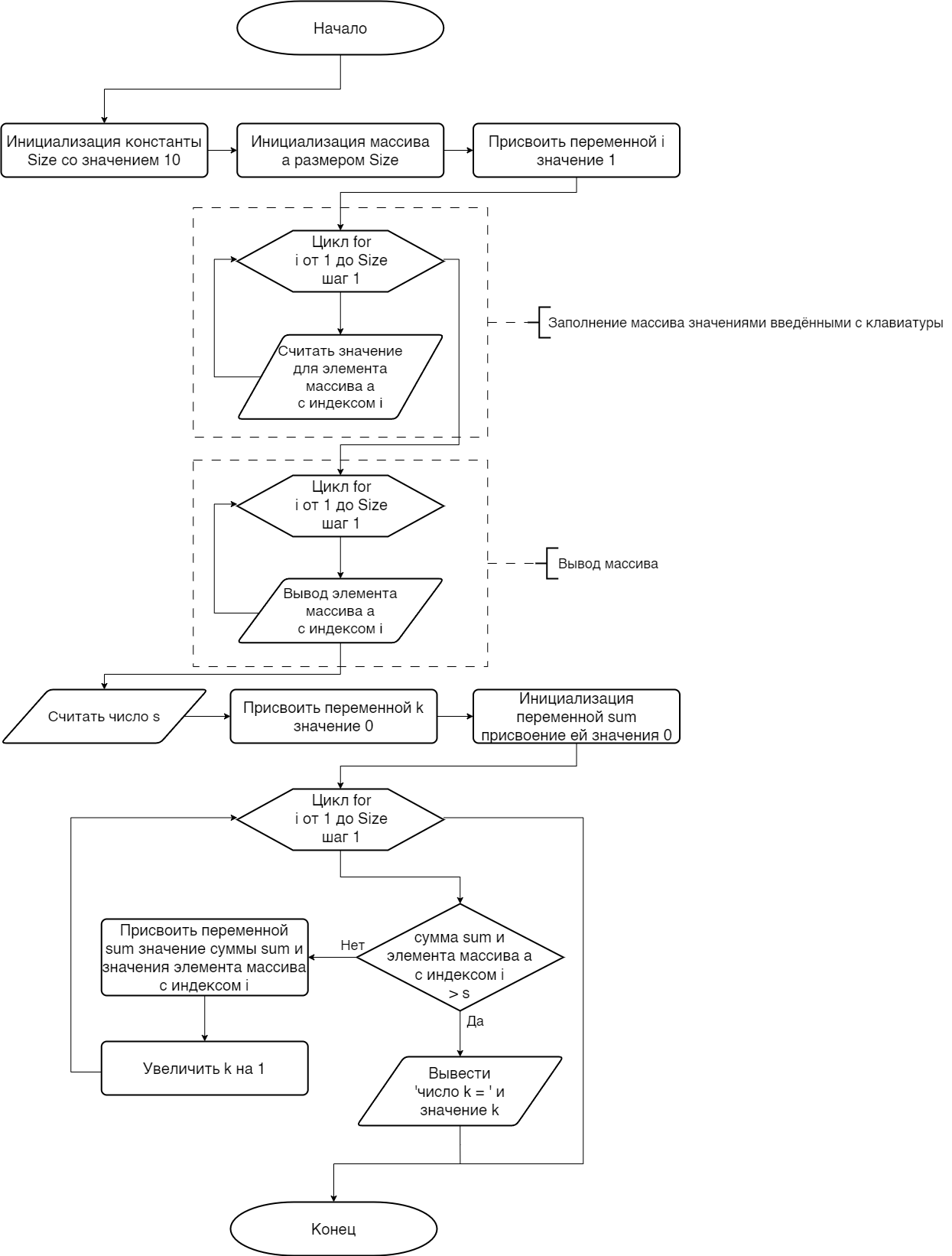
Написать программу, выполняющую поиск такого минимального k, что сумма первых k элементов массива превосходит заданное число S.

**Описание алгоритма**

Алгоритм эффективно находит минимальное количество элементов массива, необходимое для того, чтобы их сумма превысила заданное значение *s*.

1. Инициализация переменных и массива**:**
   * *a*: Массив целых чисел размером *Size* (10 элементов).
   * *i, k, s, sum*: Целочисленные переменные, где i используется для итерации, *k* для отслеживания количества элементов, *s* для хранения заданного числа, и *sum* для накопления суммы элементов массива.
2. Ввод элементов массива**:**
   * Цикл **for** запускается от 1 до *Size*. В каждой итерации пользователю предлагается ввести значение для текущего элемента массива *a[i]*.
   * После каждого ввода экран очищается функцией **ClrScr**.
3. Вывод элементов массива:
   * После заполнения массива программа выводит его элементы на экран.
4. Ввод числа *s*:
   * Программа запрашивает у пользователя ввод числа **s**, которое будет использоваться для сравнения с суммой элементов массива.
5. Поиск минимального *k*:
   * Значение *k* инициализируется нулем.
   * Цикл **for** запускается от 1 до *Size*. В каждой итерации проверяется, превышает ли текущая накопленная сумма *sum* плюс текущий элемент массива *a[i]* заданное число *s*.
   * Если условие выполняется (сумма превышает *s*), то программа увеличивает *k* на 1, выводит значение *k* и завершает выполнение с помощью команды **exit**.
   * Если условие не выполняется, программа увеличивает сумму *sum* на значение текущего элемента массива и увеличивает *k* на 1.
6. Завершение программы:
   * Если программа не находит такое *k*, при котором сумма превышает **s** (то есть проходит весь массив), она просто завершается без дополнительного вывода.

**Схема алгоритма с комментариями**

****

**Код программы**

**program** lab\_2;

**uses** crt;

**const** Size = 10;

**var**

a: **array**[1..Size] **of** Integer;

i, k, s, sum: integer;

**begin**

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

**begin**

writeln('Введите a[',i,']');

readln(a[i]);

ClrScr;

**end**;

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

write(a[i],' ');

writeln();

writeln('Введите s');

readln(s);

k:=0;

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

**begin**

**if** (sum+a[i])>s **then**

**begin**

k := k+1;

writeln('число k= ',k);

**exit**;

**end**

**else**

**begin**

sum := sum + a[i];

k:= k + 1;

**end**;

**end**;

**end**.

**Результат выполнения программы**

1 2 3 4 5 6 7 8 12 34 Введите s 10 число k = 5

3 9 27 81 729 123 434 2313 12312 23 Введите s 1000 число k = 7

0 0 0 1 0 0 0 1 0 1 Введите s 2 число k = 10

**Формулировка задания (с вариантом)**

Вариант: 1

Задание 2:

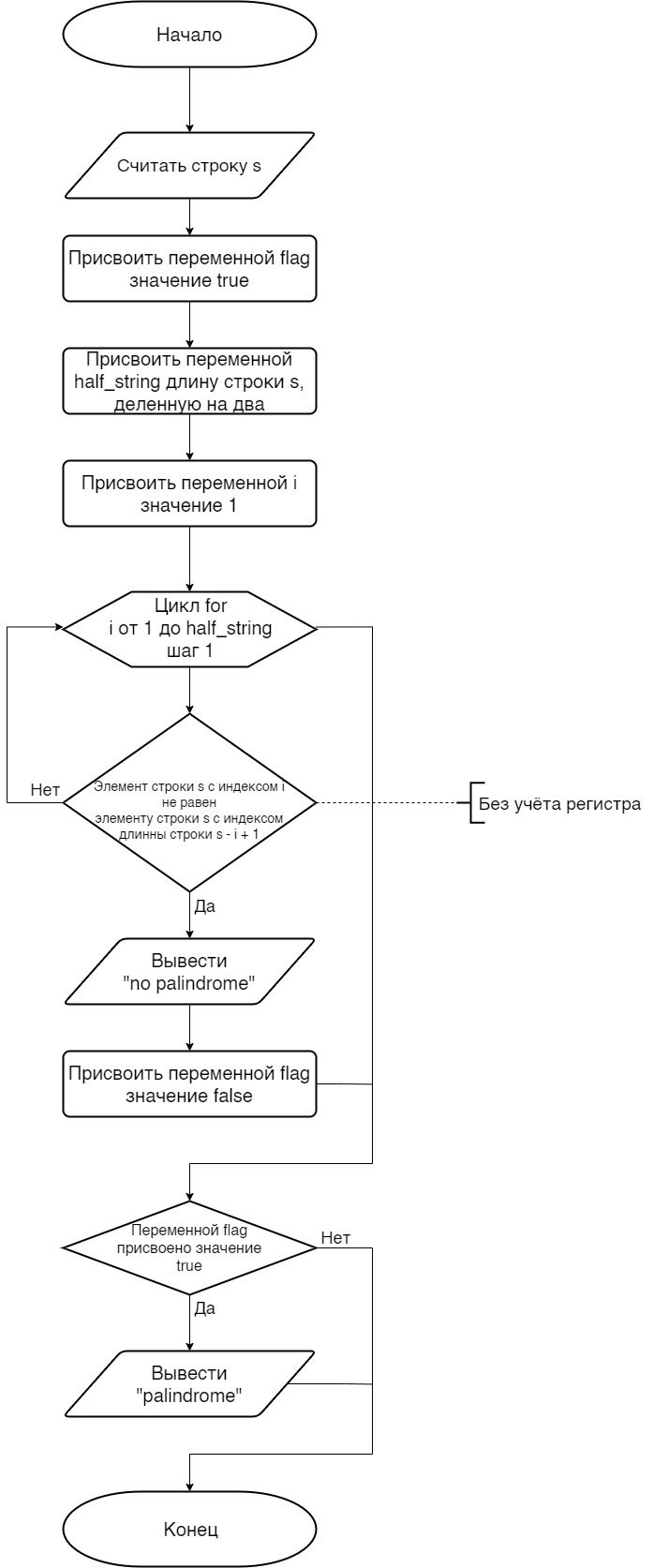
Написать программу, выполняющую поиск такого минимального k, что сумма первых k элементов массива превосходит заданное число S.

**Описание алгоритма**

Программа эффективно проверяет строки на предмет палиндрома, не учитывая регистр символов, путем сравнения соответствующих символов с противоположных концов строки. Если обнаруживается несоответствие, делается вывод, что строка не является палиндромом.

1. Объявление переменных**:**
   * *s*: Переменная строкового типа для хранения вводимой строки.
   * *i*: Переменная типа byte, используемая как счетчик в цикле.
   * *flag*: Логическая переменная, используемая как флаг для указания на то, является ли строка палиндромом.
2. Ввод данных**:**
   * Программа предлагает пользователю ввести строку, которая считывается и сохраняется в переменную *s*.
3. Проверка на палиндром:
   * Изначально *flag* устанавливается в значение **true**, что означает предположение, что строка является палиндромом.
   * Цикл **for** используется для итерации по строке. Цикл выполняется от первого символа до середины строки.
   * Внутри цикла каждый символ на позиции *i* с начала строки (**s[i]**) сравнивается с соответствующим символом с конца (**s[length(s)-i+1]**).
   * Функция Upcase используется для преобразования обоих символов в верхний регистр, что обеспечивает нечувствительность к регистру при сравнении.
   * Если какая-либо пара символов не совпадает, программа выводит сообщение 'no palindrome', устанавливает *flag* в значение **false** и выходит из цикла.
4. Вывод результатов:
   * Если цикл завершается без обнаружения несоответствия (то есть если *flag* остается равным **true**), программа выводит сообщение 'palindrome', указывая на то, что строка является палиндромом.
   * Программа ожидает дополнительный ввод (вероятно, чтобы пользователь мог увидеть результат) перед завершением.

**Схема алгоритма с комментариями**

****

**Код программы**

**program** lab\_2;

**var**

s: string;

i, half\_string: byte; flag: boolean;

**begin**

write('String: ');

readln(s);

flag := true;

half\_string := length(s) **div** 2;

**for** i:=1 **to** half\_string **do**

**if** Upcase(s[i]) <> Upcase(s[length(s)-i+1]) **then**

**begin**

writeln('no palindrome');

flag:= false;

**break**;

**end**;

**if** flag **then**

write('palindrome');

readln

**end**.

**Результат выполнения программы**

String: 12345654321

palindrome

String: абугагаагагуба

palindrome

String: нажалКАБАНнаБАКЛАЖАН

Palindrome

String: Was mich nicht umbringt, macht mich stärker

no palindrome

**Вывод**

В ходе выполнения этой лабораторной работы были созданы две программы на языке Pascal, которые решают две различные задачи: поиск минимального значения k, при котором сумма первых k элементов массива превышает заданное число S, и определение, является ли введенная строка палиндромом. Эта работа демонстрирует использование массивов и строк, а также применение циклов и условных операторов для решения задач.

В первом задании программа успешно определяет минимальное количество элементов массива, необходимых для превышения заданной суммы, что показывает понимание работы с одномерными массивами и накопления значений. Во втором задании программа эффективно проверяет строки на предмет палиндрома, демонстрируя умение работы со строками как с частным случаем одномерных массивов, а также учитывая регистр символов.

Оба задания вместе подчеркивают важность логического мышления в программировании и способность использовать основные конструкции языка для решения конкретных задач. Таким образом, выполнение этой лабораторной работы способствовало углублению знаний и навыков в области программирования на Pascal, а также позволило лучше понять принципы работы и использования массивов и строк в решении практических задач.