Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №7

Тема работы: Множества

Выполнил

студент: гр. 151004 Данилов Ф.А.

Проверил: Фадеева Е.П.

Минск 2022

Содержание

[1 Постановка задачи 4](#_Toc101650447)

[2 Описание алгоритмов решения задачи 5](#_Toc101650448)

[3 Структура данных 6](#_Toc101650449)

[3.1 Структура типов программы 6](#_Toc101650450)

[3.2 Структура данных основной программы 6](#_Toc101650451)

[3.3 Структура данных алгоритма check 6](#_Toc101650452)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc101650453)

[4.1 Схема алгоритма подпрограммы check 9](#_Toc101650454)

[5 Результаты расчетов программы 14](#_Toc101650455)

[Приложение А 15](#_Toc101650456)

[Приложение Б 18](#_Toc101650457)

# Постановка задачи

Дана непустая последовательность слов из строчных русских букв; между соседними словами – запятая, за последним словом точка. Вывести в алфавитном порядке все гласные буквы, которые не входят более чем в одно слово.

Примечание: гласные буквы – а, е, и, о, у, ы, э, ю, я;

согласные буквы – все остальные буквы, кроме й, ь, ъ;

звонкие согласные – б, в, г, д, ж, з, л, м, н, р;

глухие согласные – к, п, с, т, ф, х, ц, ч, ш, щ.

# Описание алгоритмов решения задачи

Таблица 1 – Описание алгоритмов решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п. | Наименование алгоритма | Назначение  алгоритма | Формальные параметры | Предпологаемый тип реализации |
|  | Основной алгоритм | Ввод строки и проверка правильности. Вызывает следующие подпрограммы:  check(str, res) |  |  |
|  | check(str, result) | Проверка соответстивия переданной строки формуле | str – получает от фактического параметра адрес с защитой;  result – получает от фактического параметра адрес, возвращаемый функцией параметр | Функция. res – возвращаймый параметр |

# Структура данных

## Структура типов программы

Таблица 2 – Структура типов программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| TNums | set of '1' .. '9' | Пользовательский тип-множество, содержащий возможные значения цифр |
| TSigns | set of '\*' .. '-' | Пользовательский тип-множество, определяющий возможные значения знаков |
| TLetters | set of 'a' .. 'g' | Пользовательский тип-множество, определяющий возможные значения букв |
| TCondition | (error, none, formula, formulaPlus, sign, name, int) | Пользовательский тип-перечисление, содержащий информацию о типе выражения на определенный момент |

## Структура данных основной программы

Таблица 3 – Структура данных основной программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| inputStr | string | Вводимая пользователем строка |
| isCorrect | boolean | Полученный результат проверки |

## Структура данных алгоритма check

Таблица 4 – Структура данных алгоритма check(str, result)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| str | string | Переданная на проверку строка | Формальный |
| i | integer | Итератор и индекс | Локальный |
| condition | TCondition | Тип выражения на определенный момент | Локальный |
| result | boolean | Результат проверки | Формальный |
| signs | TSigns | Множество знаков | Локальный |

Таблица 5 – Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| nums | TNums | Множество цифр | Локальный |
| letters | TLetters | Множество букв | Локальный |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

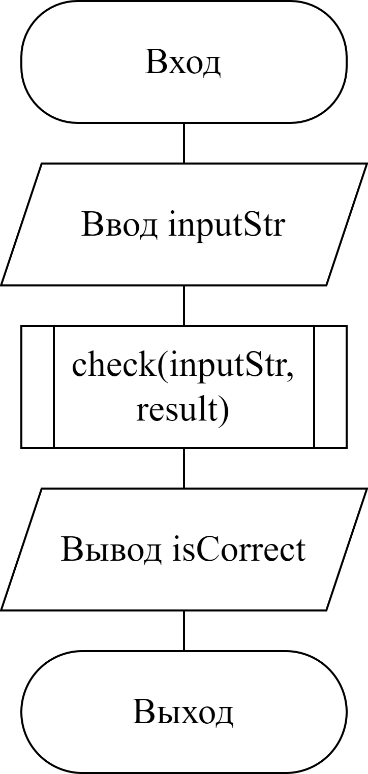


Рисунок 1 – Схема алгоритма решения задачи по ГОСТ 19.701-90

## Схема алгоритма подпрограммы check

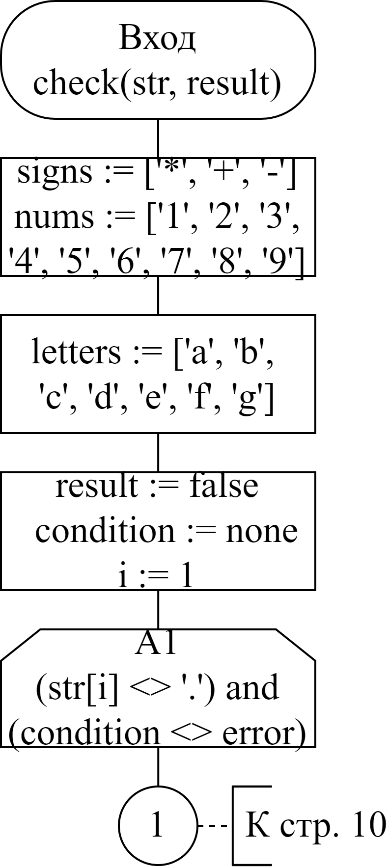


Рисунок 2 – Схема алгоритма подпрограммы check по ГОСТ 19.701-90

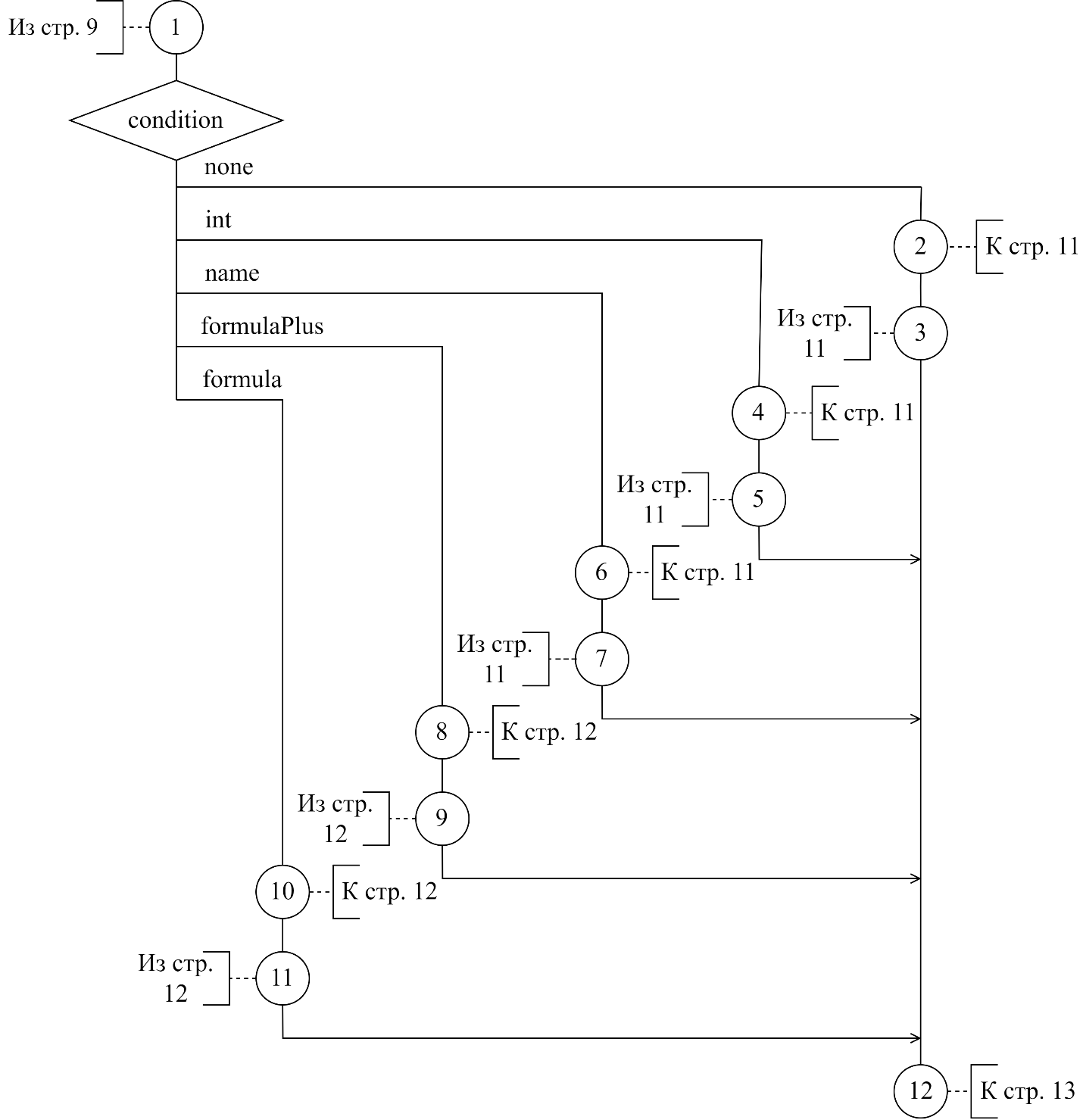


Рисунок 3 – Схема алгоритма подпрограммы check по ГОСТ 19.701-90

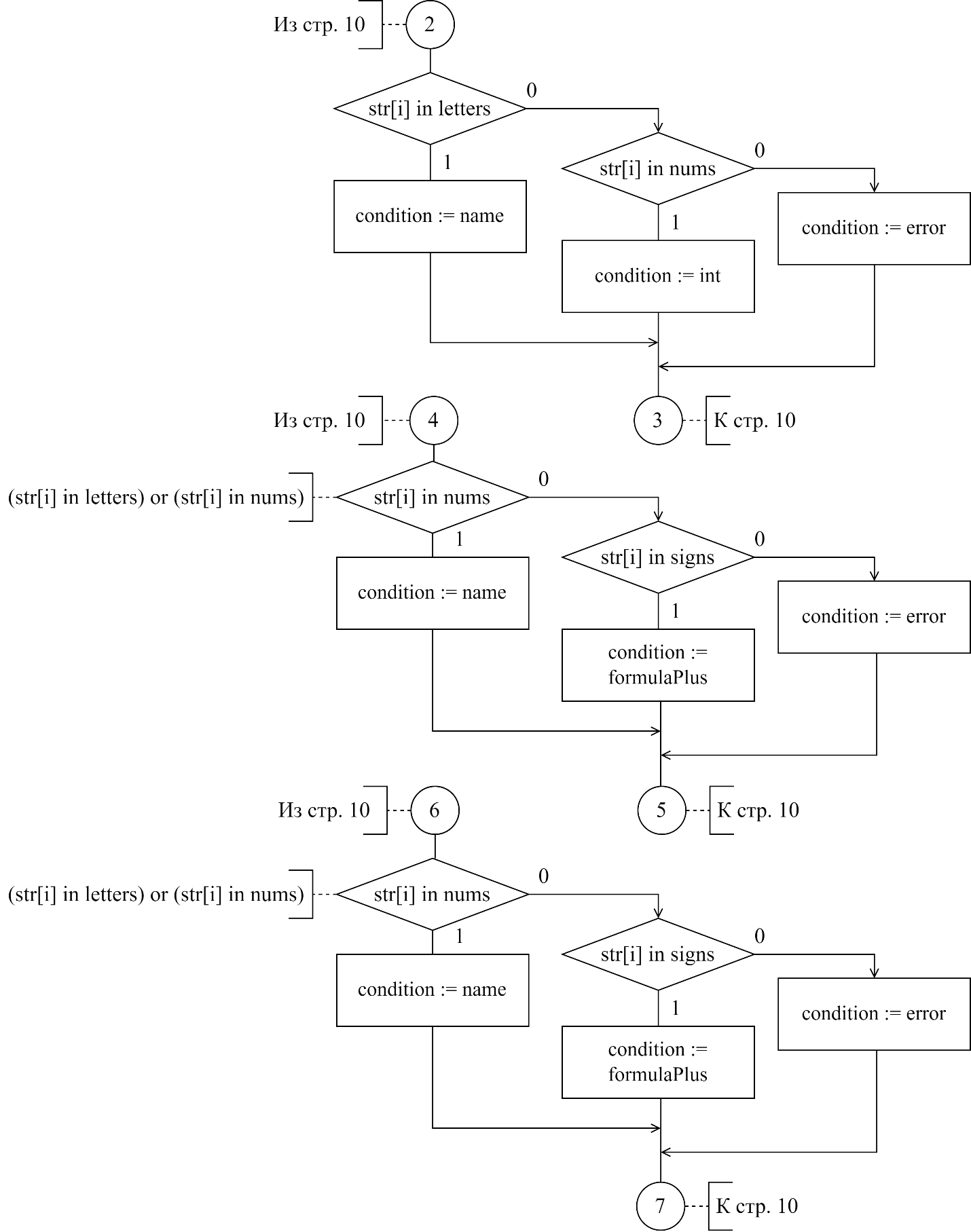


Рисунок 4 – Схема алгоритма подпрограммы check по ГОСТ 19.701-90

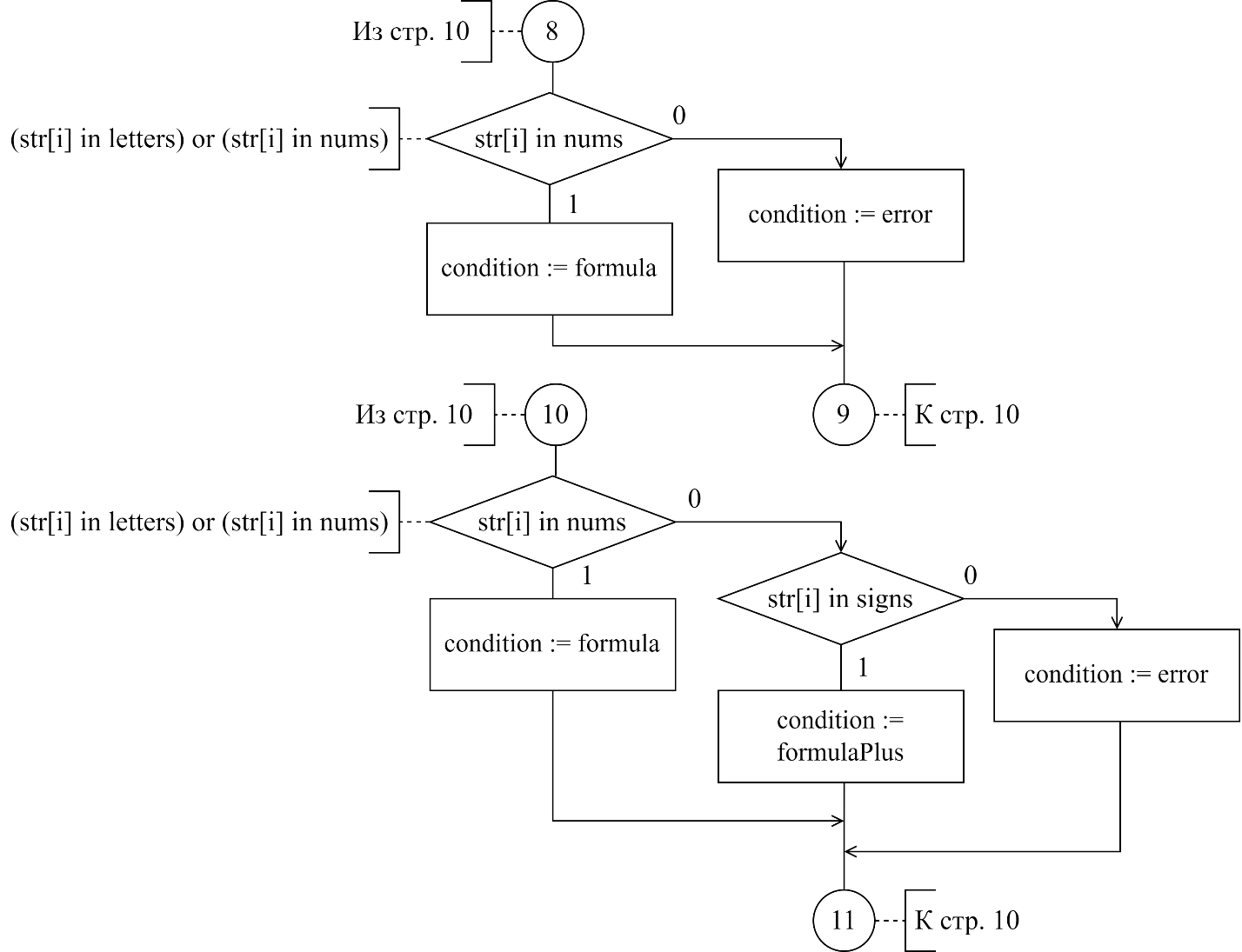


Рисунок 5 – Схема алгоритма подпрограммы check по ГОСТ 19.701-90

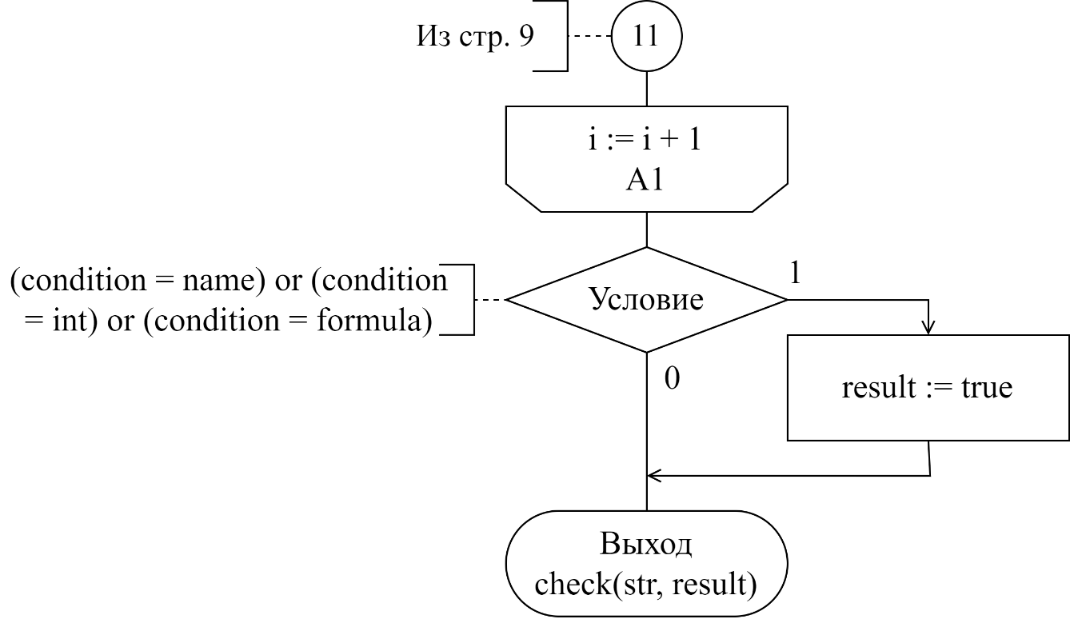


Рисунок 6 – Схема алгоритма подпрограммы check по ГОСТ 19.701-90

# Результаты расчетов программы



Рисунок 7 – Результаты расчётов программы

Приложение А

(обязательное)

Исходный код программы

program Lab7;

// A program that determines if this text is a

// valid formula

// Console app

{$APPTYPE CONSOLE}

// Modules declaration

uses

System.SysUtils;

// Vars declaration

var

inputStr: string;

{ inputStr - string the program verifies }

// A function which verifies text

function check(const str: string): boolean;

// Types declaration

type

TNums = set of '1' .. '9';

TSigns = set of '\*' .. '-';

TLetters = set of 'a' .. 'g';

TCondition = (error, none, formula, formulaPlus,

sign, name, int);

// Const declaration

const

signs: TSigns = ['\*', '+', '-'];

nums: TNums = ['1', '2', '3', '4', '5', '6', '7',

'8', '9'];

letters: TLetters = ['a', 'b', 'c', 'd', 'e',

'f', 'g'];

// Vars declaration

var

i: integer;

condition: TCondition;

begin

// Setting variables to the original values

result := false;

condition := none;

i := 1;

// Start of the cycle A1, checking every element

// and changing condition type

while (str[i] <> '.') and (condition <> error) do

begin

case condition of

// To start determining

none:

begin

if str[i] in letters then

condition := name

else if str[i] in nums then

condition := int

else

condition := error

end;

// If the condition only contains numbers

int:

begin

if str[i] in nums then

condition := int

else if str[i] in signs then

condition := formulaPlus

else

condition := error

end;

// If the condition is a name

name:

begin

if (str[i] in letters) or

(str[i] in nums) then

else if str[i] in signs then

condition := formulaPlus

else

condition := error

end;

// If the condition is a formula with something

formulaPlus:

begin

if str[i] in letters then

condition := formula

else if str[i] in nums then

condition := formula

else

condition := error;

end;

// If the condition is a finished formula

formula:

begin

if (str[i] in letters) or

(str[i] in nums) then

condition := formula

else if str[i] in signs then

condition := formulaPlus

else

condition := error

end

end;

// Finish of the cycle A1

inc(i);

end;

// The condition is a formula if it only contains

//numbers or its a name

if (condition = name) or (condition = int) or

(condition = formula) then

result := true;

end;

begin

write('Input condition: ');

readln(inputStr);

write('Is correct: ');

// Outputting the result of the verification

writeln(check(inputStr));

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы

Таблица 5 – Тестовые наборы

|  |  |  |
| --- | --- | --- |
| Тест | Исходные данные и ожидаемый результат | Полученный результат |
| 1. | Ожидается: TRUE |  |
| 2. | Ожидается: TRUE |  |
| 3. | Ожидается: TRUE |  |
| 4. | Ожидается: TRUE |  |
| 5. | Ожидается: TRUE |  |
| 6. | Ожидается: FALSE |  |
| 7. | Ожидается: FALSE |  |
| 8. | Ожидается: FALSE |  |
| 9. | Ожидается: FALSE |  |
| 10. | Ожидается: TRUE |  |
| 11. | Ожидается: FALSE |  |
| 12. | Ожидается: FALSE |  |