

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа №9

по дисциплине: Основы программирования
тема: «Использование множественного
типа»

Выполнил: ст. группы ПВ-202
Буйвало Анастасия Андреевна

Проверил:
Притчин Иван Сергеевич
Брусенцева Валентина
Станиславовна

Белгород 2020 г.

Лабораторная работа №9

«Использование множественного типа»

Цель работы: получение навыков работы с множествами.

Задания для подготовки к работе:

1. Изучите множественный тип данных языка Паскаль.
2. Разбейте задачу соответствующего варианта на подзадачи, таким образом, чтобы решение каждой подзадачи описывалось подпрограммой, а основная программа состояла бы в основном из вызовов подпрограмм.
3. Опишите блок-схему алгоритма решения задачи в укрупненных блоках.
4. Для каждой подзадачи опишите используемые структуры данных, спецификацию и блок-схему алгоритма.
5. Опишите блок-схему алгоритма решения задачи с использованием блоков «предопределённый процесс».
6. закодируйте алгоритм.
7. Подберите наборы тестовых данных с обоснованием их выбора.

Задания к работе:

1. Наберите программу, отладьте ее и протестируйте.
2. Выполните анализ ошибок, выявленных при отладке программы.

Задание варианта №7:

Дана строка. Вывести все гласные, которые есть хотя бы в одном слове.

Выполнение работы:

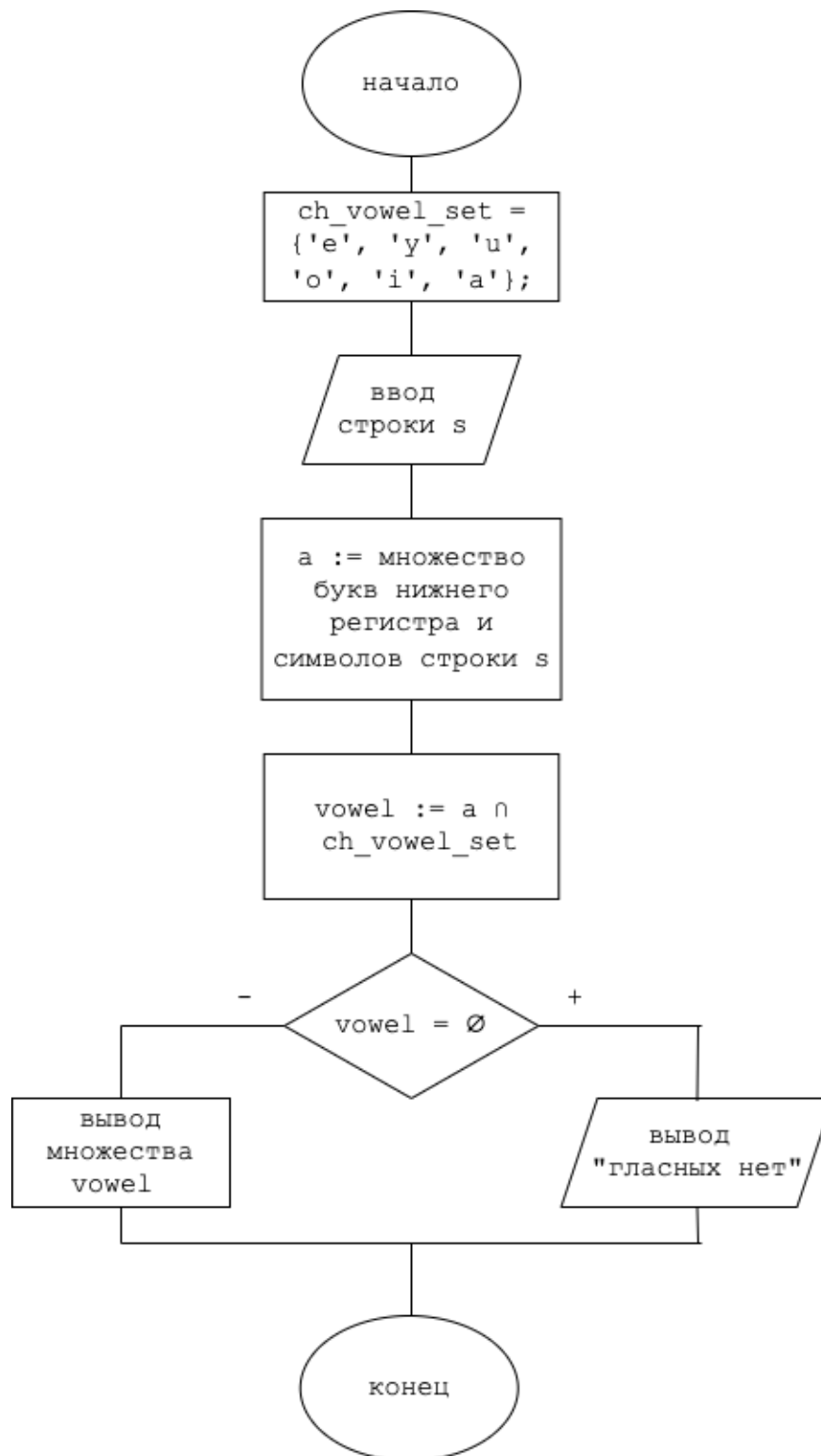
1. Выделение подзадач

Выделим следующие подзадачи:

1. Приведение гласных букв к нижнему регистру
2. Запись в множество a букв нижнего регистра и символов строки s
3. Вывод множества гласных букв, которые встречаются в строке s хотя бы один раз

Опишем алгоритм в укрупненных блоках в терминах выделенных подзадач

2. Блок-схема алгоритма в укрупненных блоках:



3. Описание структур данных:

```
const ch_vowel_set = ['e', 'y', 'u', 'o', 'i', 'a'];  
type ch_set = set of char;
```

4. Описание подпрограмм:

Процедура `input_set`

Выделим подзадачу:

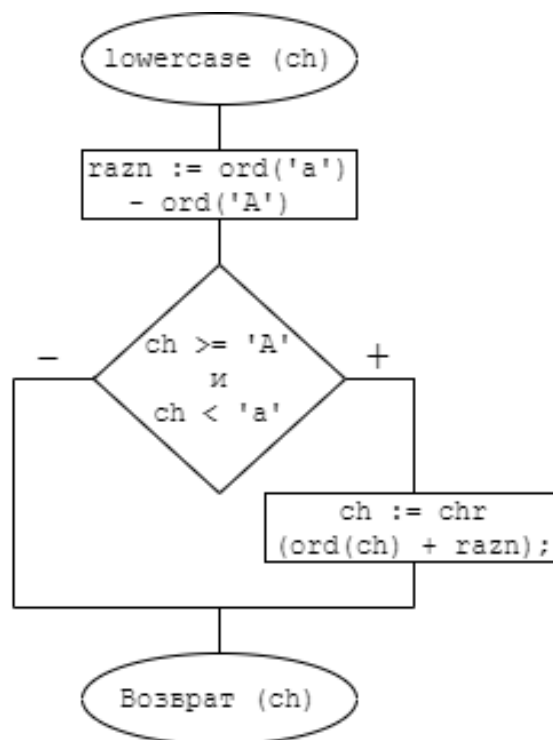
Приведение букв к нижнему регистру

Функция `lowercase`.

Спецификация.

1. Заголовок: `function lowercase (ch : char): char;`
2. Назначение: возвращает букву `ch` нижнего регистра.
3. Входные параметры: `ch`.
4. Выходные параметры: нет.

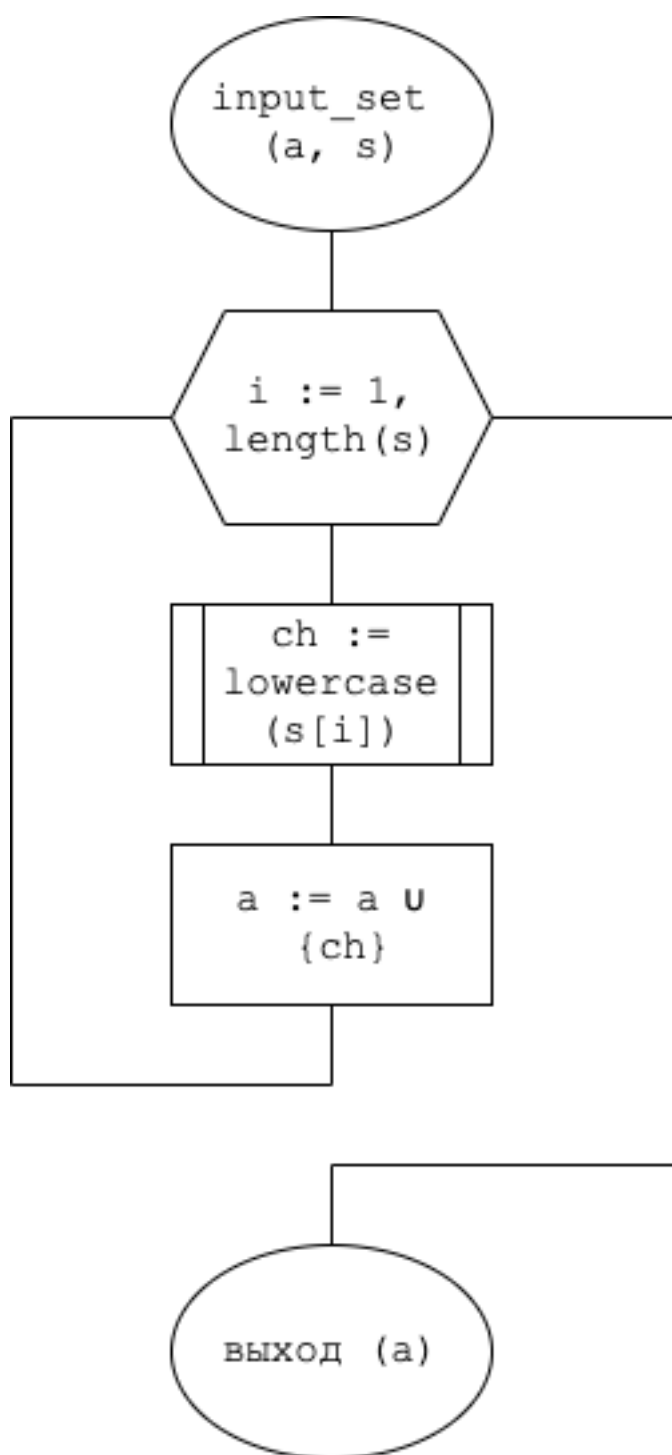
Блок-схема:



Спецификация

1. Заголовок: `procedure input_set (var a : ch_set; s: string);`
2. Назначение: Запись в множество `a` букв нижнего регистра и символов строки `s`
3. Входные параметры: `a`, `s`.
4. Выходные параметры: `a`.

Блок-схема:

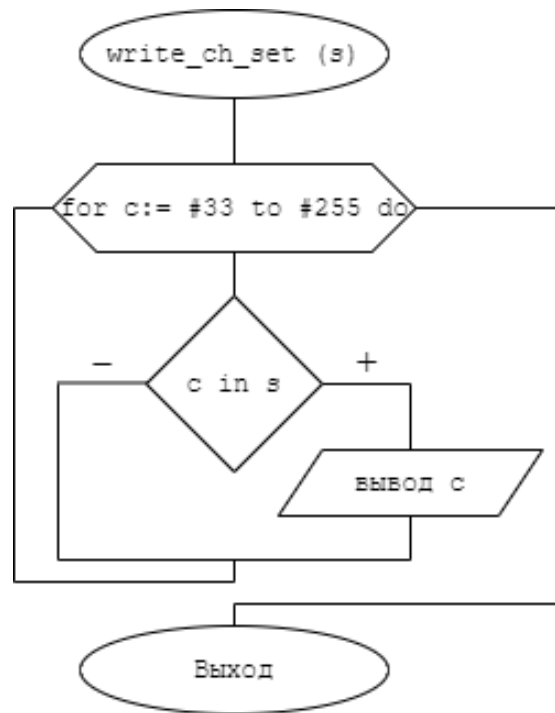


Процедура write_ch_set

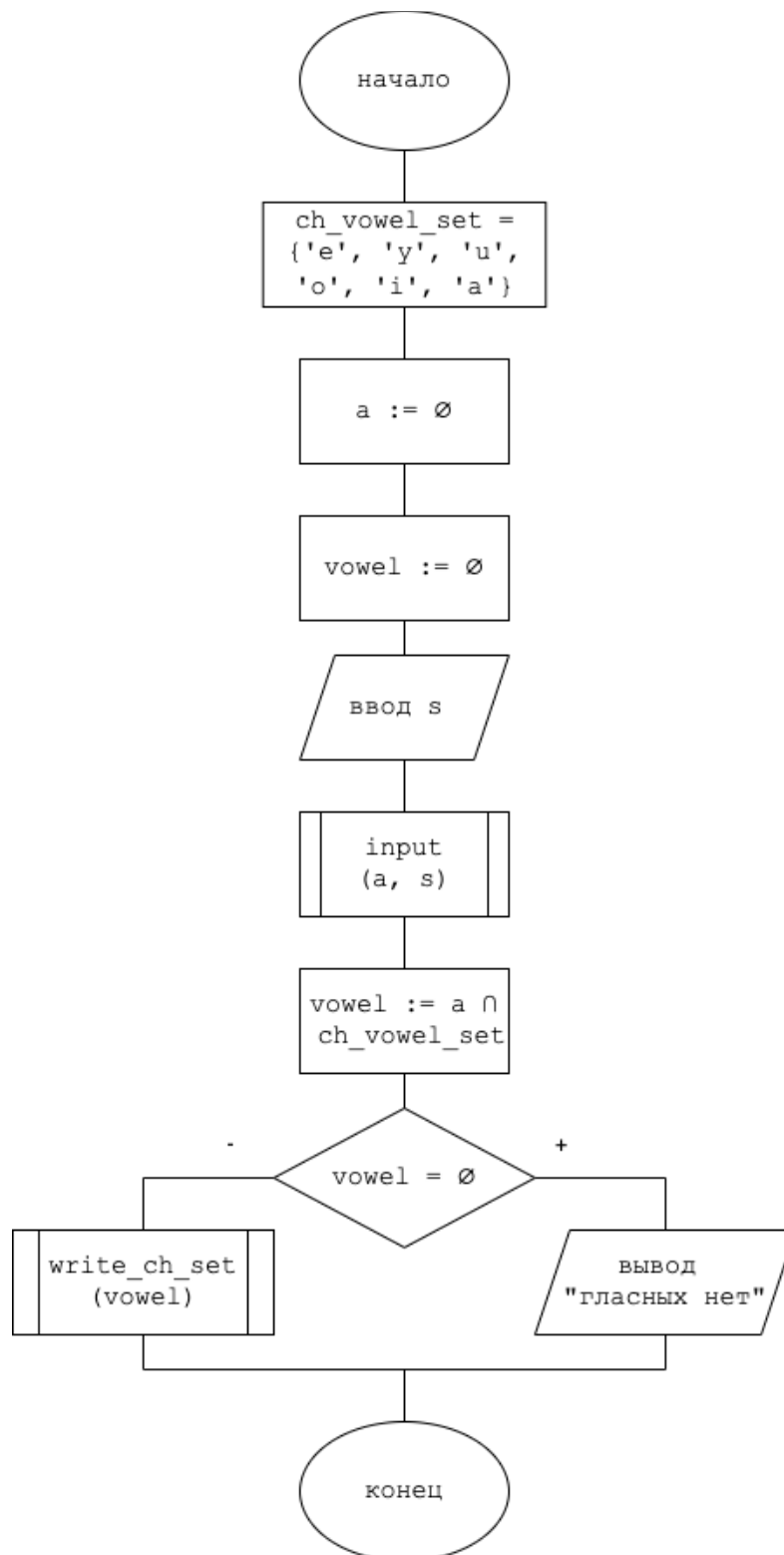
Спецификация

1. Заголовок: `procedure write_ch_set (s: ch_set);`
2. Назначение: Вывод множества `s`.
3. Входные параметры: `s`.
4. Выходные параметры: нет.

Блок-схема:



**5. Блок-схема алгоритма решения задачи с блоками
«предопределенный процесс»**



6. Тестовые данные:

| № | Исходные данные | Результаты |
|---|-----------------|-------------|
| | S | |
| 1 | nEw string | e i |
| 2 | EeliAa | a e i |
| 3 | vfskln | гласных нет |

7. Текст программы:

```
program lab9;
const ch_vowel_set = ['e', 'y', 'u', 'o', 'i', 'a'];
type ch_set = set of char;

{Возвращает букву ch нижнего регистра}
function lowercase (ch : char): char;
var razn : byte;
begin
    razn := ord('a') - ord('A');
    if (ch >= 'A') and (ch < 'a') then
        ch := chr(ord(ch) + razn);
    lowercase := ch;
end;

{Запись в множество a букв нижнего регистра и символов строки s}
procedure input_set (var a : ch_set; s : string);
var i : byte;
    ch : char;
begin
    for i := 1 to length (s) do
        begin
            ch := lowercase (s[i]);
            a := a + [ch];
        end;
    end;

{Вывод множества s}
procedure write_ch_set (s: ch_set);
var c : char;
begin
    for c := #33 to #255 do
        if (c in s) then
            write (c : 3);
    end;

var s : string;
    a, vowel : ch_set;

begin
    a := [ ];
    vowel := [ ];
    writeln ('введите строку');
    readln(s);
    input_set (a,s);
    vowel := a * ch_vowel_set;
    if vowel = [] then
        writeln('гласных нет')
    else
        begin
            write('полученная строка:');
            write_ch_set (vowel);
        end;
end;
end.
```

8. Анализ допущенных ошибок:

- При наборе программы ошибок допущено не было;

9. Результаты работы программы:

```
Окно вывода
введите строку
nEw string
полученная строка:  e  i
```

```
Окно вывода
введите строку
EeIiAa
полученная строка:  a  e  i
```

```
Окно вывода
введите строку
vfskln
гласных нет
```