Московский государственный технический

университет им. Н.Э. Баумана

Курс «Основы информатики»

Отчет по домашнему заданию

«Работа со строками»

Проверил:

Преподаватель каф. ИУ5

Аксёнова М.В.

Подпись и дата:

Факультет «Информатика и система управления»

Кафедра ИУ5 «Система обработки информации и управления»

Выполнил:

Студент группы ИУ5-13Б

Пермяков Дмитрий Кириллович

Подпись и дата:

**Постановка задачи**

11)Слова текста из малых латинских букв записаны не менее чем через один пробел; текст оканчивается точкой. БЕЗ ИСПОЛЬЗОВАНИЯ конструкции STRING:

а) написать программу ввода такого текста с клавиатуры;

б) напечатать все слова, отличающиеся от последнего слова, и совпадающие с конечным отрезком алфавита (z, yz, xyz и т.д.).

**Разработка алгоритма**

**Описание используемых переменных:**

#define SIZE 150- размер массивов

size\_t ch- символ с клавиатуры, i- кол-во введённых символов.

int arr[SIZE]- массив, хранящий введённые символы в типе <int>.

size\_t j – индекс последнего символа в массиве arr.

size\_t l- индекс на первую букву последнего слова.

size\_t k – длина последнего слова.

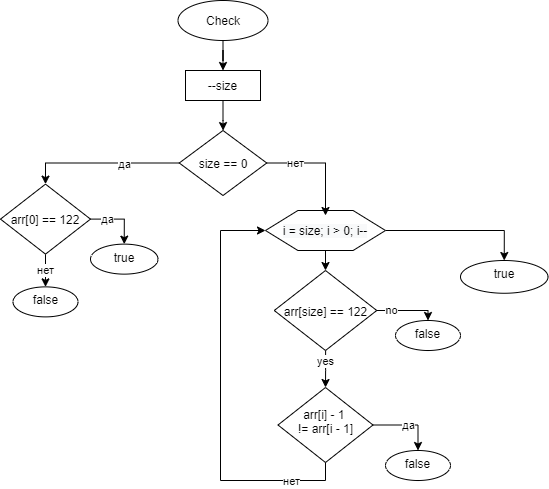
int arr2[SIZE]- массив, в который по очереди закидываем отдельное слово и передаём в функцию Check для проверки условия задачи.

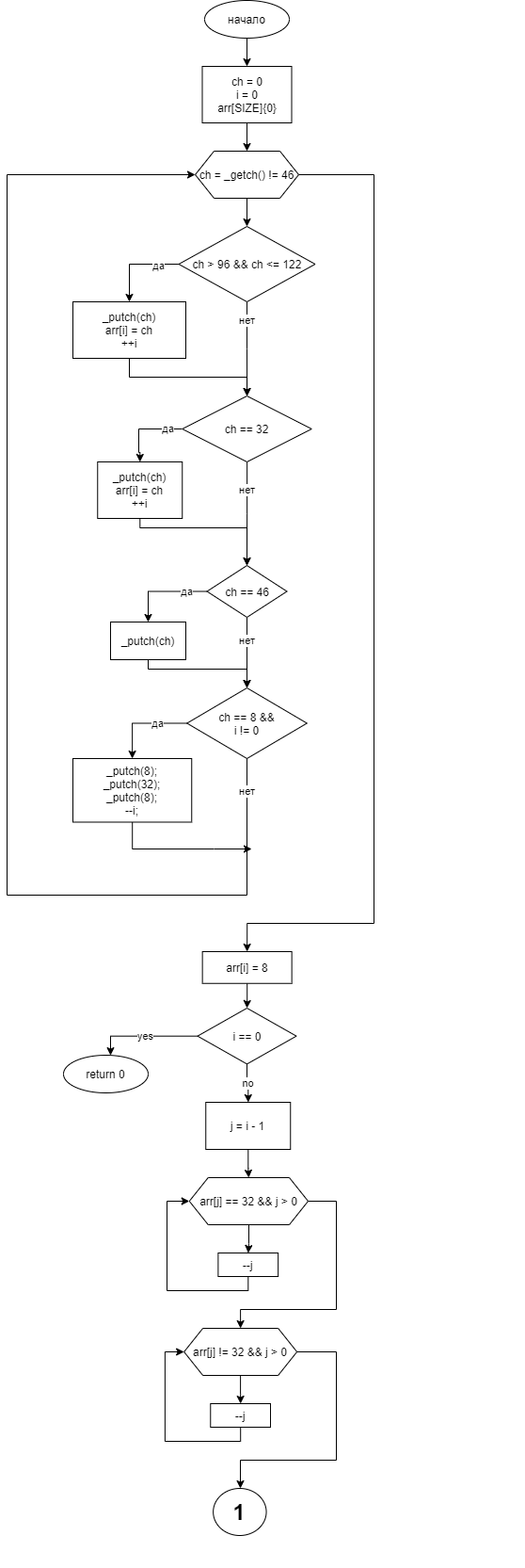
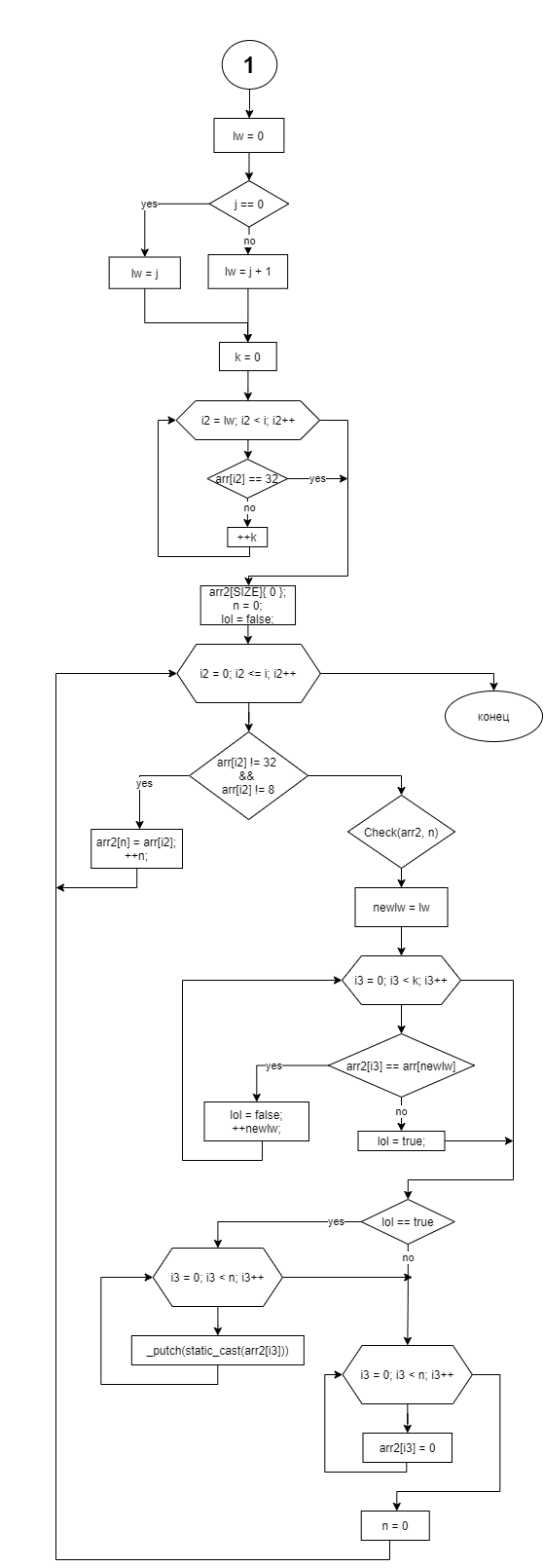
bool lol- определяет, схоже ли слово, удовлетворяющее условию задачи, с последним словом.

size\_t n- длина проверяемого слова, а также вспомогательный индекс для заполнения arr2.

**Блок схемы:**

**bool Check(int arr[SIZE], size\_t size)**



**Код программы:**

#include <conio.h>

#define SIZE 150

bool Check(int arr[SIZE], size\_t size)

{

--size;

if (size == 0)

{

if (arr[0] == 122)

return true;

else

return false;

}

for (size\_t i = size; i > 0; i--)

{

// если последняя буква z. Далее

if (arr[size] == 122)

{

// если след.буква не сходится с алфавитом. выход

if (arr[i] - 1 != arr[i - 1])

return false;

}

else

return false;

}

return true;

}

int main()

{

size\_t ch = 0, i = 0;

int arr[SIZE]{ 0 };

do

{

ch = \_getch();

// считываем только латинские буквы

if (ch > 96 && ch <= 122)

{

\_putch(ch);

arr[i] = ch;

++i;

}

// считываем пробел

if (ch == 32)

{

\_putch(ch);

arr[i] = ch;

++i;

}

// считываем точку

if (ch == 46)

\_putch(ch);

// создаем удаление

if (ch == 8 && i != 0)

{

\_putch(8);

\_putch(32);

\_putch(8);

--i;

}

} while (ch != 46);

arr[i] = 8;

if (i == 0)

return 0;

\_putch('\n');

// ищем индекс последнего слова

size\_t j = i - 1;

while (arr[j] == 32 && j > 0)

--j;

while (arr[j] != 32 && j > 0)

--j;

size\_t lw = 0;

if (j == 0)

lw = j;

else

lw = j + 1;

// кек

// размер ласт слова

size\_t k = 0;

for (size\_t i2 = lw; i2 < i; i2++)

{

if (arr[i2] == 32)

break;

++k;

}

// кек

int arr2[SIZE]{ 0 };

bool lol = false;

size\_t n = 0;

\_putch('\n');

for (size\_t i2 = 0; i2 < i; i2++)

{

// длина проверяемого слова

if (arr[i2] != 32 && arr[i2] != 8)

{

arr2[n] = arr[i2];

++n;

}

else

{

if (Check(arr2, n))

{

// сравниваем с последним словом

size\_t newlw = lw;

for (size\_t i3 = 0; i3 < k; ++i3)

{

// если буква схожа на букву ласт слова, проверяем дальше

if (arr2[i3] == arr[newlw])

{

lol = false;

++newlw;

}

// если не совпала хотя бы 1 буква. стоп

else

{

lol = true;

break;

}

// если длина слова равна длине ласт слова. стоп

if (k == n)

break;

}

// если слово != последнему. печатаем

if (lol)

{

for (size\_t i3 = 0; i3 < n; i3++)

\_putch(static\_cast<char>(arr2[i3]));

\_putch(' ');

}

}

// очищаем массив

for (size\_t i3 = 0; i3 < n; i3++)

arr2[i3] = 0;

n = 0;

}

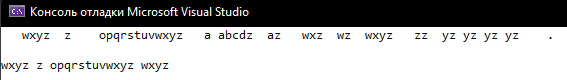
}

\_putch('\n');

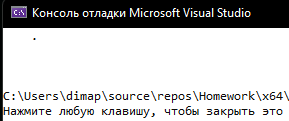
return 0;

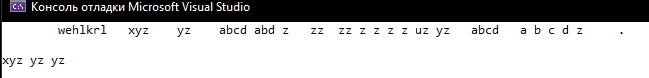
}

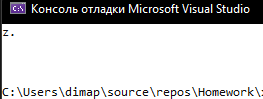
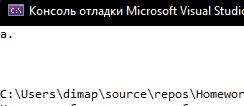
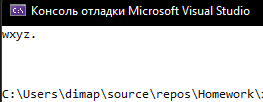
**Результат программы:**

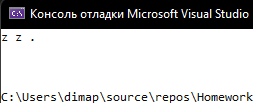
****

****

****

****

**  **

****