

**Звіт з лабораторної роботи  
за дисципліною "програмування"  
студента групи ПА-17-1  
Панасенка Егора Сергійовича  
Кафедра комп'ютерних технологій, фпм, дну  
2017/2018 навч.р.**

1. Постановка задачі: Розробіть функцію, яка визначає в рядку позицію першого символу слова, номер якого заданий.
2. Опис ходу розв'язку:
  1. Задається функція `ispartofword`, яка перевіряє чи може цей символ створювати слово, тобто йде перевірка чи є символ буквою чи цифрою або апострофом.
  2. Задається функція `find_word_number`, яка шукає позицію потрібного слова за допомогою таких дій:
    1. Якщо задали знайти перше слово і перший символ створює слово, то вивести позицію 1, а якщо ні то іти далі.
    2. Виконується цикл до тих пір поки не закінчиться рядок або ми не знайдемо потрібне слово, у циклі кожне нове слово це символ, який створює слово і якийсь інший символ.
    3. Якщо ми знайшли слово то вивести позицію слово, якщо ні, то вивести помилку.
  3. Задається функція `ign_other`, у якій будуть ігноруватися усі данні до кінця рядка для того щоб непотрібні дані не заважали в майбутньому.
  4. Задається функція `get_string`, яка забирає рядок з консолі або з файлу та зберігає його у масив з типом `wchar_t` який дозволяє вміщувати у собі делілько-байтні символи як один символ, а розмір рядка обмежується максимально доступним значенням `int`, а також розмір збільшується частинами рівними `CHUNK (128)`.
  5. У головній функції йде запит на рядок доки не отримаємо рядок, який можна опрацювати, запит на потрібне слово та вихід позиції слова.

### 3. Вихідний текст програми розв'язку задачі

```
#include <stdio.h>
#include <stdlib.h>
#include <wchar.h>
#include <locale.h>
#include <wctype.h>
#include <limits.h>
#define CHUNK 128

int ispartofword(wchar_t c) {
    return iswalphac(c) || iswalnum(c) || c == '\\';
}

int find_word_number(wchar_t * string, int word_num) {
    int x = 0, i = 0;
    if (ispartofword(string[0])) x = 1;
    for (i = 1; string[i] && x < word_num; i++)
        if (!ispartofword(string[i-1]) && ispartofword(string[i]))
            x++;
    if (x == word_num)
        return i;
    else
        return -1;
}

void ign_other(FILE * input) {
    wchar_t c = 0;
    while ((c = fgetwc(input)) != EOF && c != '\\n') {}
}

wchar_t * get_string (FILE * input, int * status) {
    wchar_t * a = (wchar_t *) calloc(CHUNK, sizeof(wchar_t)), ch = EOF;
    int length = 0;
    while (ch) {
        ch = fgetwc(input);
        if (length >= INT_MAX - 2 * CHUNK || ch == EOF || ch == '\\n') {
            if (ch == EOF)
                *status = 2;
            else if (ch == '\\n')
                *status = 3;
            else if (length >= INT_MAX - 2 * CHUNK)
                *status = 1;
            ch = 0;
        }
        if (length % CHUNK == CHUNK - 1)
            a = (wchar_t *) realloc (a, (length + CHUNK + 1) *
sizeof(wchar_t));
        a[length++] = ch;
    }
    return a;
}

int main() {
    setlocale(LC_ALL, "");
    int word_num = 0, found = -1, status = 0, i, mw = 0;
    wchar_t * a;
    while (status < 2) {
```

```

wprintf(L"Введите строку: ");
a = get_string(stdin,&status);
if (status == 1) {
    system("clear");
    wprintf(L"Слишком длинная строка, максимальная длина строки %i.
Попробуйте ещё.\n", INT_MAX - 2 * CHUNK);
    ign_other(stdin);
} else if (status > 1) {
    if (ispartofword(a[0])) mw++;
    if (a[0])
        for (i=1; a[i]; i++)
            if (!ispartofword(a[i-1]) && ispartofword(a[i]))
                mw++;
    if (mw == 0) {
        status = 0;
        system("clear");
        wprintf(L"Эта строка пустая\n",INT_MAX);
    }
}
if (status < 2) free(a);
}
while (found == -1) {
    wprintf(L"Введите номер слова: ");
    wscanf(L"%i",&word_num);
    ign_other(stdin);
    if (word_num <= mw)
        found = find_word_number(a,word_num);
    if (found == -1) {
        system("clear");
        wprintf(L"Нет столько слов сколько вы просите. Максимальное
количество слов: %i.\n",mw);
    }
}
wprintf(L"Слово находится на %i позиции\n",found);
return 0;
}

```

#### 4. Опис інтерфейсу програми:

1. Запит рядка який буде містити букви та цифри та буде містити не більше 2147483391 символів.
2. Якщо ці умови не виконуються, то програма буде запитувати рядок доки не отримає задовільний рядок.
3. Запит номера слова який потрібно знайти.
4. Якщо стільки слів немає у рядку, то програма буде запитувати число доки не отримає число менше ніж слів у рядку
5. Вихід позиції слова

#### 5. Опис тестових прикладів:

Введите строку: a sddf asdf sdfa  
Введите номер слова: 3  
Слово находится на 8 позиции

Введите строку: Тестовое сообщение для поиска  
Введите номер слова: 3  
Слово находится на 23 позиции