

Міністерством освіти та науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт

Про виконання лабораторної роботи № 9

«Стрічки»

виконав

Студент групи ФЕП-11с

Гарбіч Данило

Перевірено:

Асистент Кужій Ю. І.

Львів 2024 р.

2. Написати програму, яка реалізує введення із консолі прислів'я або афоризму із 7+ слів однією стрічкою, розбиває її на масив окремих слів із зворотнім порядком символів в словах і виводить їх в окремих рядках.
3. Написати програму, яка реалізує пострічкове введення із консолі списку Вашої групи (улюбленої футбольної команди, будь-якого колективу з 10+ учасників) в форматі Номер-[пробіл]-Прізвище-[пробіл]-Ім'я, а потім 1) підраховує і виводить кількість відмінних (різних) імен та середню довжину прізвищ; 2) формує і виводить в алфавітному порядку список виду Номер-[пробіл]-Ім'я-[пробіл]-Прізвище.
4. Написати програму, яка перевіряє на правильність введену із консолі адресу електронної пошти. Правильний формат: префікс@домен1. домен2 [.домен3]. Приклади: 1) правильні - johndeer@gmail.com, ivan.franko@lnu.edu.ua; 2) неправильні – jackdoe@gmail, Іван2020@укр.нет.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

void reverse_word(char* word) {
    int len = strlen(word);
    for (int i = 0; i < len / 2; i++) {
        char temp = word[i];
        word[i] = word[len - i - 1];
        word[len - i - 1] = temp;
    }
}

int validate_email(const char* email) {
    const char* at = strchr(email, '@');
    const char* dot = strchr(email, '.');

    return at && dot && at < dot && at != email && *(dot + 1) != '\0';
}

void task1_reverse_proverb() {
    char input[256];
    printf("Enter a proverb or aphorism with 7+ words: ");
    fgets(input, sizeof(input), stdin);

    char* word = strtok(input, " ");
    printf("\nReversed words:\n");
    while (word != NULL) {
        reverse_word(word);
        printf("%s ", word);
        word = strtok(NULL, " ");
    }
    printf("\n");
}

void task2_process_participants() {
    int numbers[100];
    char surnames[100][50];
    char names[100][50];
    char unique_names[100][50];
    int count = 0, unique_count = 0, total_length = 0;
    char buffer[100];

    printf("\nEnter participants in the format Number Surname Name (type 'end' to finish):\n");

    while (1) {
        fgets(buffer, sizeof(buffer), stdin);
        if (strcmp(buffer, "end", 3) == 0) break;

        sscanf(buffer, "%d %s %s", &numbers[count], surnames[count], names[count]);
        total_length += strlen(surnames[count]);
        count++;
    }

    for (int i = 0; i < count; i++) {
        int is_unique = 1;
        for (int j = 0; j < unique_count; j++) {
            if (strcmp(unique_names[j], names[i]) == 0) {
                is_unique = 0;
                break;
            }
        }
        if (is_unique) {
            strcpy(unique_names[unique_count], names[i]);
            unique_count++;
        }
    }

    double avg_length = (double)total_length / count;

    for (int i = 0; i < count - 1; i++) {
        for (int j = i + 1; j < count; j++) {
            if (strcmp(surnames[i], surnames[j]) > 0) {
                int temp_num = numbers[i];
                numbers[i] = numbers[j];
                numbers[j] = temp_num;

                char temp_surname[50];
                strcpy(temp_surname, surnames[i]);
                strcpy(surnames[i], surnames[j]);
                strcpy(surnames[j], temp_surname);

                char temp_name[50];
                strcpy(temp_name, names[i]);
                strcpy(names[i], names[j]);
                strcpy(names[j], temp_name);
            }
        }
    }

    printf("\nNumber of unique names: %d\n", unique_count);
    printf("Average surname length: %.2f\n", avg_length);
    printf("List in format Number-Name-Surname:\n");

    for (int i = 0; i < count; i++) {
        printf("%d %s %s\n", numbers[i], names[i], surnames[i]);
    }
}

void task3_validate_email() {
    char email[256];
    printf("\nEnter an email address: ");
    fgets(email, sizeof(email), stdin);
    email[strcspn(email, "\n")] = 0;

    if (validate_email(email)) {
        printf("Email address is valid.\n");
    } else {
        printf("Email address is invalid.\n");
    }
}

int main() {
    task1_reverse_proverb();
    task2_process_participants();
    task3_validate_email();
}

```