

Лабораторна робота № 22

Робота з рядками

Мета: отримання практичних навиків роботи з рядками.

Завдання 1:

10	Перевернути (представити у зворотному порядку) кожне слово у рядку, що складається з парної кількості літер.
----	--

Листинг програми:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int* reversed(char(*words)[100], char* string) {
    char* pointer = strtok(string, " ");
    int index = 0;
    while (pointer != NULL) {
        strcpy(words[index], pointer);
        pointer = strtok(NULL, " ");
        index++;
    }
    for (int j = 0; j < index; j++) { if (strlen(words[j]) % 2 == 0) {
        _strrev(words[j]); } }
    words[index][0] = '\0';
    for (int index = 0; strlen(words[index]) > 0; index++) { printf("%s ",
words[index]); }
    return 0;}
int main() {
    char string[100], words[100][100];
    printf("Enter a string: "); fgets(string, sizeof(string), stdin);
    if (string[strlen(string) - 1] == '\n') { string[strlen(string) - 1] = '\0'; }
    reversed(words, string);
    return 0;} return 0;}
```

Enter a string: is that something to do with the lack of cookies
si taht something ot od htiw the kcal fo cookies

Рис. 1 (Результат виконання програми Завдання 1)

Завдання 2:

Завдання 2. Дано рядок зі стандартним набором символів-роздільників між словами. Необхідно:

- 1) перевірити, чи є у рядку слова паліндроми;
- 2) визначити кількість повторень кожного слова;
- 3) знайти слово, що повторюється в рядку максимальну кількість разів;

					ДУ«Житомирська політехніка».23.121.10.000 – Лр22						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Семенчук О.А.			Звіт з лабораторної роботи			Лім.	Арк.	Аркушів	
Перевір.		Чижмотря О. В								1	5
Керівник								ФІКТ Гр. ІПЗ-23-1[2]			
Н. контр.											
Зав. каф.											

- 4) видалити із рядка повторення слів, сформувати рядок із різних слів;
- 5) переставити слова в алфавітному порядку;

Листинг програми:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <limits.h>
void token(char words[100][100], char* string) {
    char* pointer = strtok(string, " ");
    int index = 0;
    while (pointer != NULL) {
        strcpy(words[index], pointer);
        pointer = strtok(NULL, " ");
        index++;
    }
}
void palindrome(char words[100][100], char palindromearray[100][100]) {
    int count = 0;
    for (int index = 0; strlen(words[index]) > 0; index++) {
        if (strcmp(words[index], _strrev(_strdup(words[index]))) == 0 &&
            strlen(words[index]) != 1) {
            strcpy(palindromearray[count], words[index]);
            count++;
        }
        if (strlen(palindromearray[0]) > 0) {
            printf("Palidromes : \n");
            for (int index = 0; strlen(palindromearray[index]) > 0; index++) {
                printf("%s ", palindromearray[index]);
            }
        }
        else { printf("The string does not have palidromes!"); }
    }
}
void withoutrepeatedwords(char words[100][100], char countarray[100][100]) {
    int count = 0;
    for (int index = 0; strlen(words[index]) > 0; index++) {
        bool found = false;
        for (int j = 0; strlen(countarray[j]) > 0; j++) {
            if (strcmp(words[index], countarray[j]) == 0) {
                found = true;
                break;
            }
        }
        if (!found) {
            strcpy(countarray[count], words[index]);
            count++;
        }
    }
    printf("\nArray without reptead words : \n");
    for (int index = 0; strlen(countarray[index]) > 0; index++) { printf("%s ", count-
array[index]); }
}
void countinstances(char words[100][100], char countarray[100][100], char countint[100])
{
    for (int index = 0; strlen(countarray[index]) > 0; index++) { countint[index] = 0;
}
    for (int index = 0; strlen(countarray[index]) > 0; index++) {
        for (int j = 0; strlen(words[j]) > 0; j++) {
            if (strcmp(countarray[index], words[j]) == 0) {
                countint[index] = countint[index] + 1;
            }
        }
    }
    printf("\nWords and number of their instances : \n");
    for (int index = 0; strlen(countarray[index]) > 0; index++) { printf("%s[%d] ",
countarray[index], countint[index]); }
}
void wordthatoccursthemost(char countarray[100][100], char countint[100]) {
    int max = INT_MIN;
    int maxindex;
    for (int index = 0; strlen(countarray[index]) > 0; index++) {
        if (countint[index] > max) {
            max = countint[index];
            maxindex = index;
        }
    }
    printf("\nWord that occurs the most : ");
    printf("%s", countarray[maxindex]);
}
void alphabeticalorder(char words[100][100]) {
    int length = 0;
}
```

		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр22	Арк.
		Чижмоторя О. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

while (strlen(words[length]) > 0) { length++; }
for (int index = 0; index < length - 1; index++) {
    for (int j = 0; j < length - index - 1; j++) {
        if (strcmp(words[j], words[j + 1]) > 0) {
            char temp[100];
            strcpy(temp, words[j]);
            strcpy(words[j], words[j + 1]);
            strcpy(words[j + 1], temp);}}}
printf("\nArray sorted in alphabetical order : \n");
for (int index = 0; strlen(words[index]) > 0; index++) { printf("%s ",
words[index]); }
int main() {
    char string[100], words[100][100], palindromearray[100][100], countar-
ray[100][100], countint[100];
    printf(">"); fgets(string, sizeof(string), stdin);
    if (string[strlen(string) - 1] == '\n') { string[strlen(string) - 1] = '\0'; }
    token(words, string);
    palindrome(words, palindromearray);
    withoutrepeatedwords(words, countarray);
    countinstances(words, countarray, countint);
    wordthatoccursthemost(countarray, countint);
    printf("\nArray without reptead words : \n");
    for (int index = 0; strlen(countarray[index]) > 0; index++) { printf("%s ", count-
array[index]); }
    alphabeticalorder(words);
    return 0;}

```

Результат виконання програми:

```

>aga atom aga
Palidromes :
aga aga
Words and number of their instances :
aga[2] atom[1]
Word that occurs the most : aga
Array without reptead words :
aga atom
Array sorted in alphabetical order :
aga aga atom

```

Рис. 2 (Результат виконання програми Завдання 2)

		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр22	Арк.
		Чижевотря О. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

Завдання на самостійну роботу:

(а) З використанням оператора switch перетворюватиме ціле число (від 2 до 5), що вводиться з клавіатури, у відповідний запис («незадовільно», «задовільно», «добре», «відмінно»), в разі введення іншого числа – виводити повідомлення про відсутність такої оцінки:

Листинг програми:

```
#include <stdio.h>
void grade(char character) {
    switch (character) {
        case '2': printf("unsatisfactory"); break;
        case '3': printf("satisfactory"); break;
        case '4': printf("good"); break;
        case '5': printf("excellent"); break;
        default: printf("Please, enter a valie grade!");}}
int main() {
    printf("Enter a grade : "); char character; scanf_s("%c", &character);
    grade(character);
    return 0;}
```

Результат виконання програми:

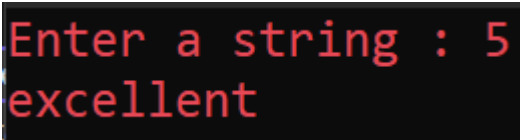


Рис 3. (Результат виконання програми самостійної роботи (а))

(b) Переписати програму із пункту а без використання оператора switch, але з використанням масиву рядків:

Листинг програми:

```
#include <stdio.h>
void grade(char character) {
    switch (character) {
        case '2': printf("unsatisfactory"); break;
        case '3': printf("satisfactory"); break;
        case '4': printf("good"); break;
        case '5': printf("excellent"); break;
        default: printf("Please, enter a valie grade!");}}
int main() {
    printf("Enter a grade : "); char character; scanf_s("%c", &character);
    grade(character);
    return 0;}
```

Результат виконання програми:

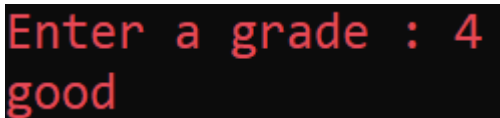


Рис 4. (Результат виконання програми самостійної роботи (b))

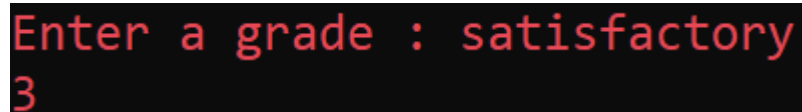
		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр22	Арк.
		Чижмоторя О. В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

(с) Написати програму, яка буде здійснювати зворотне перетворення до завдання № b, тобто перетворювати введену з клавіатури оцінку у вигляді рядка тексту у числове значення:

Листинг програми:

```
#include <stdio.h>
#include <string.h>
void grades(char result[100]) {
    char grades[4][100] = { "unsatisfactory", "satisfactory", "good", "excellent" };
    bool found = false;
    for (int index = 0; strlen(grades[index]) > 0; index++) {
        if (strcmp(result, grades[index]) == 0) {
            printf("%d", index + 2);
            found = true;
        }
    }
    if (!found) printf("Please, enter a valid grade!");
}
int main() {
    printf("Enter a grade : ");
    char result[100];
    scanf_s("%s", result, sizeof(result));
    grades(result);
    return 0;}
```

Результат виконання програми:



```
Enter a grade : satisfactory
3
```

Рис 5. (Результат виконання програми самостійної роботи (с))

Github link : <https://github.com/FearlessAtom/Lab22>

Висновок : Під час виконання лабораторної роботи було здобуто практичні навички роботи з рядками у мові програмування C, що дозволить зручно та ефективно використовувати маніпуляції рядками у подальших програмних проектах.

		Семенчук О. А.			ДУ «Житомирська політехніка».23.121.10.000 – Лр22	Арк.
		Чижомотря О. В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		