

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

до лабораторної роботи №3

із дисципліни «Програмування»

на тему

ОПРАЦЮВАННЯ СИМВОЛЬНИХ ДАНИХ

Виконав:

студент групи КМ-93

Пиндиківський Т. Р.

Керівник:

асистент Дрозденко О. М.

ЗМІСТ

МЕТА РОБОТИ	3
ПОСТАНОВКА ЗАДАЧІ	4
ОСНОВНА ЧАСТИНА	5
ВИСНОВКИ	7
ДОДАТОК А	8
ДОДАТОК Б	10

МЕТА РОБОТИ

Вивчити опис символічних даних та операцій над ними.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1:

1. Дано текст. Визначити, чи є він паліндромом, тобто чи читається од-наково як зліва направо, так і справа наліво. Різницю малих і великих літер до уваги не брати.

ОСНОВНА ЧАСТИНА

Завдання 1 :

Для виконання програми спочатку імпортуються наступні модулі :

- ***stdio.h*** – файл заголовку для стандартних операцій введення/виведення;
- ***string.h*** – бібліотека мови C, що містить функціонал для роботи з рядковими величинами;
- ***stdbool.h*** – бібліотека, що дозволяє використовувати логічний тип даних **bool** у мові C.

Змінні, що використовуються під час виконання завдання:

- ***sentence[1000]*** (*char*) – рядкова величина, в якій зберігається значення введеного користувачем вислову;
- ***length*** (*int*) – цілочисельна величина, що дорівнює кількості букв вислову;
- ***i*** (*int*) – допоміжне число для ітерації по елементах вислову;
- ***result*** (*bool*) - логічне значення, що визначає, є введена рядкова величина паліндромом, чи ні;

Хід виконання завдання:

1. Вводиться значення рядка ***sentence***.
2. Оскільки в умові задачі ігнорується регістр символів вислову, то далі в циклі ***for*** відбувається переведення кожного символу у нижній регістр з використанням функції ***tolower()***.
3. Підраховується кількість символів вислову функцією ***strlen()***.
4. За допомогою циклу ***for*** відбувається перевірка рівності символів, що знаходяться на однакових «відстанях» від початку та кінця вислову. Якщо знайдено хоча б одну пару символів, що не дорівнюють один одному, змінна ***result*** набуває значення ***false*** та відбувається вихід з циклу з оголошенням повідомлення у консоль, що введене користувачем речення

не є паліндромом. Якщо ж усі пари містять однакові елементи, то змінна *result* набуває значення *true*, що свідчить про те, що число є паліндромом.

ВИСНОВКИ

На цій лабораторній роботі було вивчено опис символічних даних та операцій над ними.

ДОДАТОК А

1. Програмна реалізація задачі №1

```
#include<stdio.h>
#include<stdbool.h>
#include<string.h>
void program1 ()
{
    char sentence[1000]="";
    int i=0;

    printf("\n\nEnter the sentence you want to check on being a palindrom
statement : ");
    gets(sentence);

    int length, j=0;

    length=strlen(sentence);

    for(i=0;i<length;i++)
    {
        if(isalpha(sentence[i]) || isdigit(sentence[i]))
        {
            j++;
        }
    }
    char sentence2[j];
    for(i=0, j=0;i<length;i++)
    {
        if(isalpha(sentence[i]) || isdigit(sentence[i]))
        {
            sentence2[j]=sentence[i];
            j++;
        }
    }

    if(!(isalpha(sentence2[strlen(sentence2)-1]) ||
isdigit(sentence2[strlen(sentence2)-1])))
        sentence2[strlen(sentence2)-1]='\0';

    printf("\n\nYour sentence \"%s\" is ", sentence);

    length=strlen(sentence2);

    for(i=0;i<length;i++)
        sentence2[i]=tolower(sentence2[i]);

    bool result;
    result=true;

    for(i=0; i<=(length/2);i++)
        if(!(sentence2[i]==sentence2[length-1-i]))
        {
            result=false;
            break;
        }
    printf(result ? "a palindrom." : "not a palindrom.");
}
```


Taras Pyndykivskiy
Laboratory work #3
14th variant

Task 1 : Text is given. Determine if it is a palindrome.
The difference between lowercase and uppercase letters is ignored.

Press the number of task you want to test (1) : 1

Task 1 : Text is given. Determine if it is a palindrome.
The difference between lowercase and uppercase letters is ignored.

Enter the sentence you want to check on being a palindrom statement : Pulup hah pULUP

Your sentence Pulup hah pULUP is a palindrom.

If you want to continue testing program, press c button ... ☐

Рис.1 – Тестування завдання №1

ДОДАТОК Б

Відповіді на запитання:

1. Як описують рядки мовою C ?

Рядки в мові C - масиви символів, які завершуються нуль-символом '\0'.

2. Функції для введення та виведення символів.

Функція для введення : `scanf("%c", char_value);`

Функція для виведення : `printf("%c", char_value);`

3. Функції для введення та виведення рядків.

Функція для введення : `scanf("%s", string_value), gets(string_value);`

Функція для виведення : `printf("%s", string_value), puts(string_value);`

4. Функції перевірки символів

Функції з бібліотеки повертають значення «істина», якщо:

`isalpha(c)`: c — символ алфавіту;

`isupper(c)`: c — символ верхнього регістру;

`islower(c)`: c — символ нижнього регістра;

`isdigit(c)`: c — цифра від 0 до 9;

`isxdigit(c)`: c — шістнадцяткова цифра;

`isalnum(c)`: c — буква чи цифра;

`isspace(c)`: c — символ пробілу, табуляції, переведення рядка чи формату.

5. Функції, що реалізують операції з рядками.

Функція	Опис
<code>char* strcat(char* s1, char* s2)</code>	приєднує s2 до s1, повертає s1
<code>char* strncat(char* s1, char* s2, int n)</code>	приєднує не більше за n символів s2 до s1, завершує рядок символом '\0', повертає s1
<code>char* strcpy(char* s1, char* s2)</code>	копіює s2 в s1, включаючи '\0', повертає s1
<code>char* strncpy(char* s1, char* s2, int n)</code>	копіює не більше за n символів s2 в s1, повертає s1
<code>int strcmp(char* s1, char* s2)</code>	порівнює s1 і s2, повертає значення 0, якщо рядки еквівалентні
<code>int strncmp(char* s1, char* s2, int n)</code>	порівнює початкові n символів s1 і s2, повертає значення 0, якщо початкові n символів рядків еквівалентні
<code>int strlen(char* s)</code>	повертає кількість символів в s
<code>char* strset(char* s, char c)</code>	заповнює s символами, код яких дорівнює c, повертає вказівник на s
<code>char* strnset(char* s, char c, int n)</code>	замінює перші n символів s символами, код яких дорівнює c, повертає вказівник на s

6. З'єднання послідовностей символів.

Для з'єднання послідовностей символів використовується функція `strcat(value1, value2);`

7. Пошук першого входження символу в рядок.

Використовується функція `strchr(whole_string, value_to_find);`

8. Порівняння рядків.

Для порівняння рядків використовується функція `int strcmp(value1, value2).`

9. Копіювання символів.

Для копіювання символів використовується функція `strncpy(value1, value2, number_of_symbols);`

10.Визначення довжини рядка.

Для визначення довжини рядка використовується функція
`strlen(string_value);`

11.Скільки байтів буде виділено під розміщення масиву a в результа-ті такого оголошення: `char a[] = "ABCD";`

Буде виділено 4 байти.

12.У чому різниця в оголошеннях `char b = "Array of char"` і `char b[] = "Array of char"`?*

Різниця полягає у тому, що у першому випадку оголошується вказівник на рядкову величину, а у другому – сама рядкова величина.

13.Наведіть приклад оголошення змінної, якій можна присвоїти результат обчислення виразу 'a'?

`Char symbol='a';`