

Завдання. Задано текстовий рядок: **"The quick brown Fox jumps over a lazy Dog"**.

Розробити програму, яка виводить на екран довжину кожного із слів текстового рядка, використовуючи при цьому типові керуючі структури на асемблері. У програмі передбачити виведення на екран заданого текстового рядка і кінцевого результату.

Перш ніж приступити до написання програми, потрібно спочатку вирішити, яким чином розв'язати проблему, тобто поставлене завдання. Розв'язання проблеми можна умовно поділити на кілька кроків:

- розуміння проблеми;
- визначення шляхів розв'язання проблеми в довільній формі;
- представлення розв'язання проблеми в комп'ютерно-орієнтованій формі;
- переклад розв'язання на одну із мов програмування;
- тестування і відлагодження програми.

Розуміння проблеми. Розуміння призначення програми зазвичай включає такі моменти: вхідні дані, обробка, вихідні дані (Input-Processing-Output або IPO).

Вхідні дані. В нашому завданні є текстовий рядок, який повинен бути описаний у програмі відповідною рядковою змінною (масив символів).

Обробка. Обробка вхідних даних полягає в тому, що в текстовому рядку перевіряються по черзі всі символи, шукається кінець кожного слова і підраховується кількість букв у цьому слові.

Вихідні дані. Після визначення кінця слова його довжина виводиться на екран і обнуляється, після чого обробка масиву символів продовжується.

Розв'язання проблеми. Для роботи з масивом доцільно виділити індексну змінну. Ще одна змінна цілого типу може бути використана як лічильник кількості літер у поточному слові. Для визначення довжини слова можна використовувати символ пропуску, який поділяє речення на окремі слова (ASCII-код 20h або 32). Для спрощення програми вважаємо, що речення починається із букви і закінчується буквою, слова речення розділені лише одним символом пропуску.

Словесний опис алгоритму роботи програми може бути такий:

Взяти перший символ рядка. Якщо це буква, збільшити лічильник букв і перейти до наступного символу. Якщо це пропуск – на екран виводиться значення лічильника букв, після чого він обнуляється і обробка масиву продовжується. Якщо виявлено останню букву – виводиться значення лічильника букв і програма закінчується.

Представлення алгоритму розв'язання проблеми. Для представлення розв'язання проблеми в комп'ютерно-орієнтованій формі використовують блок-схеми (чи діаграми діяльності UML) або алгоритмічну мову (псевдокод).

I варіант.

Задати рядок "The quick brown Fox jumps over a lazy Dog".

Визначити довжину рядка.

Ініціалізувати індекс масиву та лічильник букв.

До тих пір, поки рядок не закінчився, повторювати такі операції:

Якщо поточний символ – пропуск або рядок закінчився, друкувати довжину слова і обнулити лічильник букв, інакше перейти на наступний символ і збільшити лічильник букв.

II варіант.

```
myStr = "The quick brown Fox jumps over a lazy Dog"
strLen = lengthof (myStr)
index = 0
charCount = 0
WHILE index ≤ strLen
    if (myStr[index] = ' ') OR (index = strLen)
    {
        print charCount
        charCount = 0
    }
    increment index
    increment charCount
END // while
```

Реалізація алгоритму мовою C:

```
#include <stdio.h>
#include <string.h>

int main()
{
    char mystr[] = "The quick brown Fox jumps over a lazy Dog";

    int index = 0;
    int wordlen = 0;
    int len = strlen(mystr);

    printf ("%s\n", mystr);
    while (index <= len)
    {
        if (mystr[index] == ' ' || index == len)
        {
            printf ("wordlenth = %d\n", wordlen);
            wordlen = 0;
        }
        wordlen++;
        index++;
    }
    return 0;
}
```

Реалізація алгоритму мовою Асемблер:

```
include asmio16.inc
.data
mystr db "The quick brown Fox jumps over a lazy Dog",0
strlen=$-mystr-1
wordlen dw 0

.code
.begin
    lea dx, mystr
    call WriteString
    call Crlf
    xor si, si          ; index = 0
    mov cx, strlen
cycle:                  ; for (cx=strlen; cx>0; cx--)
    mov al, mystr[si]   ; mystr[index] -> al
    cmp al, ' '         ; if al = Space
    je print            ; jump to print
    inc wordlen          ; wordlen++
    cmp cx, 1           ; if end of string
    je print            ; jump to print
    jmp short goon
print:
    mov ax, wordlen
    call WriteInt
    call Crlf
    mov wordlen, 0      ; wordlen = 0
goon: inc si            ; next symbol
    loop cycle
.end
```

Тестування і відлагодження програми. Наведені вище програми мовами С та Асемблер потрібно перевірити на працездатність і правильність функціонування. Для цього потрібно визначити довжину кожного слова рядка вручну і порівняти з результатами, які видає програма. При необхідності програму потрібно відлагодити. Для цього використовують вбудовані засоби відлагодження середовища розробки програм або відлагоджувальники сторонніх виробників, зокрема TurboDebugger від Borland для програм, написаних мовою асемблера.