

Міністерство освіти і науки, молоді та спорту України  
Національний університет “Львівська політехніка”

Кафедра ЕОМ



## Звіт

з лабораторної роботи №1 з дисципліни:  
“Системного програмування”

на тему: «Ознайомлення з середовищем MASM32. Структура  
програми на асемблері. Програмування вводу та виводу.»

Варіант: № 24.

Виконав:  
ст. групи КІ-203  
Ширий Б. І.  
Перевірила:  
стар вик. кафедри ЕОМ  
Ногаль М. В.

## ЗАВДАННЯ ВИВОДУ ПРИЗВИЩА:

### ВИБІР СЕРЕДОВИЩА:

Для виконання завдань поставлених у цій лабораторній роботі використовував Visual Studio 2022, а також перевіряв виконання на MASM32. Відповідно, створив пустий C++ файл, де змінив «Залежності збірки» та додав до проекту файл з розширенням .asm, У властивостях цього файлу змінивши тип на «Microsoft Macro Assembler».

### УМОВА ЗАВДАННЯ:

За допомогою текстового редактора створити файл <прізвище>.asm , який містить програму, приведену на лістингу 1. (Параметр <прізвище> повинен містити Ваше прізвище в англійській транслітерації і мати довжину до 8 символів). В тексті програми змінити повідомлення, що міститься за міткою HelloMessage так, щоб воно відображало Ваше прізвище. Зберегти внесені зміни. Створити <прізвище>.exe-файл засобами системи MASM32 або Visual Studio 2019.

Лістинг 1. Текст з методичних вказівок.

```
.686
.model flat, stdcall
option casemap:none
include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
includelib \masm32\lib\kernel32.lib

.data
hConsoleOutput dd 0
NumberOfCharsWritten dd 0
HelloMessage db 'Hello, world', 10, 13
NumberOfCharsToWrite dd $-HelloMessage

.code
start:
push -11
call GetStdHandle
mov hConsoleOutput, eax

push 0
push offset NumberOfCharsWritten
push NumberOfCharsToWrite
push offset HelloMessage
push hConsoleOutput
call WriteConsoleA

push 0
call ExitProcess

end start
```

## ВИКОНАННЯ:

Отож, налаштувавши Visual Studio, змінив текст, наведений у методичних вказівках, так, щоб за міткою HelloMessage повідомлення відображало моє прізвище, а саме “Shyryi”. Відповідно, початковий текст наведений у лістингу 1, а змінений у лістингу 2.

Лістинг 2. Змінений текст повідомлення Shyryi

```
.686
.model flat, stdcall
option casemap:none
include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
includelib \masm32\lib\kernel32.lib
.data
hConsoleOutput dd 0
NumberOfCharsWritten dd 0
ShyryiMessage db 'Shyryi', 10, 13
NumberOfCharsToWrite dd $-ShyryiMessage
.code
start:
push -11
call GetStdHandle
mov hConsoleOutput, eax
push 0
push offset NumberOfCharsWritten
push NumberOfCharsToWrite
push offset ShyryiMessage
push hConsoleOutput
call WriteConsoleA
push 0
call ExitProcess
end start
```

Створивши файл з розширенням .exe, запустив програму щоб перевірити її справність та отримав відповідний вивід, зображений на рисунку 1.

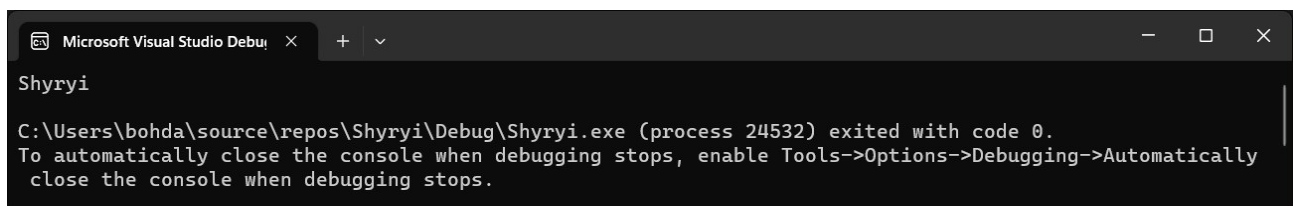


Рисунок 1. Вивід програми виводу прізвища.

Отож, текст програми відредагований правильно.

# ЗАВДАННЯ ВИВОДУ ЗА ДОПОМОГОЮ ПСЕВДОГРАФІКИ

## УМОВА ЗАВДАННЯ:

Створити програму, яка в текстовому режимі виводить за допомогою псевдографіки зображення заданого варіантом символу. Перевірити роботу створеної програми і продемонструвати її виконання.

Для 24 варіанту потрібно вивести символ: “И”

## ВИКОНАННЯ:

Проаналізувавши приклад з методички, виконав завдання написавши текст зображений на лістингу 3.

Лістинг 3. Текст виконаного завдання виводу за допомогою псевдографіки.

```
.686
.model flat, stdcall
option casemap:none
include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
includelib \masm32\lib\kernel32.lib

.data
hConsoleOutput dd 0
NumberOfCharsWritten dd 0
Symbol db 60 dup (32), 10, 13
db 1 dup (32), 8 dup (219), 20 dup (32), 8 dup (219), 10, 13
db 2 dup (32), 6 dup (219), 22 dup (32), 6 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 22 dup (32), 6 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 21 dup (32), 7 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 20 dup (32), 3 dup (219), 1 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 19 dup (32), 3 dup (219), 2 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 18 dup (32), 3 dup (219), 3 dup (219), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 17 dup (32), 3 dup (219), 4 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 16 dup (32), 3 dup (219), 5 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 15 dup (32), 3 dup (219), 6 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 14 dup (32), 3 dup (219), 7 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 13 dup (32), 3 dup (219), 8 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 12 dup (32), 3 dup (219), 9 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 11 dup (32), 3 dup (219), 10 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 10 dup (32), 3 dup (219), 11 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 9 dup (32), 3 dup (219), 12 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 8 dup (32), 3 dup (219), 13 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 7 dup (32), 3 dup (219), 14 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 6 dup (32), 3 dup (219), 15 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 5 dup (32), 3 dup (219), 16 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 4 dup (32), 3 dup (219), 17 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 3 dup (32), 3 dup (219), 18 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 2 dup (32), 3 dup (219), 19 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 4 dup (219), 1 dup (32), 3 dup (219), 20 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 7 dup (219), 21 dup (32), 4 dup (219), 10, 13
db 3 dup (32), 6 dup (219), 22 dup (32), 4 dup (219), 10, 13
db 2 dup (32), 6 dup (219), 22 dup (32), 6 dup (219), 10, 13
db 1 dup (32), 8 dup (219), 20 dup (32), 8 dup (219), 10, 13
NumberOfCharsToWrite dd $-Symbol
ReadBuf db 128 dup (?)
```

```

hConsoleInput dd 0

.code
start:
call AllocConsole
push -11
call GetStdHandle
mov hConsoleOutput, eax
push 0
push offset NumberOfCharsWritten
push NumberOfCharsToWrite
push offset Symbol
push hConsoleOutput
call WriteConsoleA

push -10
call GetStdHandle
mov hConsoleInput, eax
push 0
push offset NumberOfCharsWritten
push 128
push offset ReadBuf
push hConsoleInput
call ReadConsoleA

push 0
call ExitProcess
end start

```

Запустивши програму, отримав вивід, який зображений на рис 2, нижче.

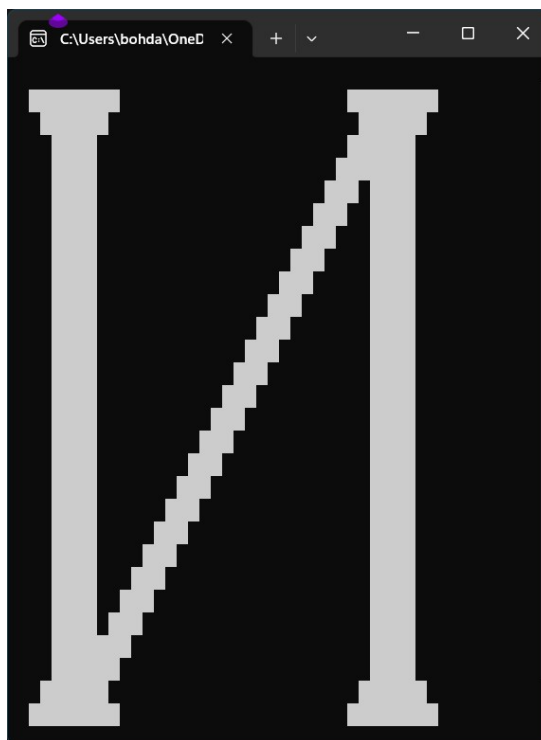


Рисунок 2. Вивід програми другого завдання.

## ВИСНОВКИ:

Освоїв послідовність дій для компіляції програм, написаних на мові Assembler за допомогою середовищ MASM32 і Visual Studio, а також навчився реалізовувати ввід з клавіатури та вивід на екран символічних даних.