### Experiment Guidebook 3

#### Experimental requirements and objective

1. Be able to code, assemble, and execute a program with Visual Studio and MASM.
2. Know how to link your programs to an external code library.
3. Know how to create procedure using assembly language.

#### Experimental environment

1. Hardware environment

The microcomputer CPU more than Pentium, more than 120GB capacity hard drive, more than 1GB of memory.

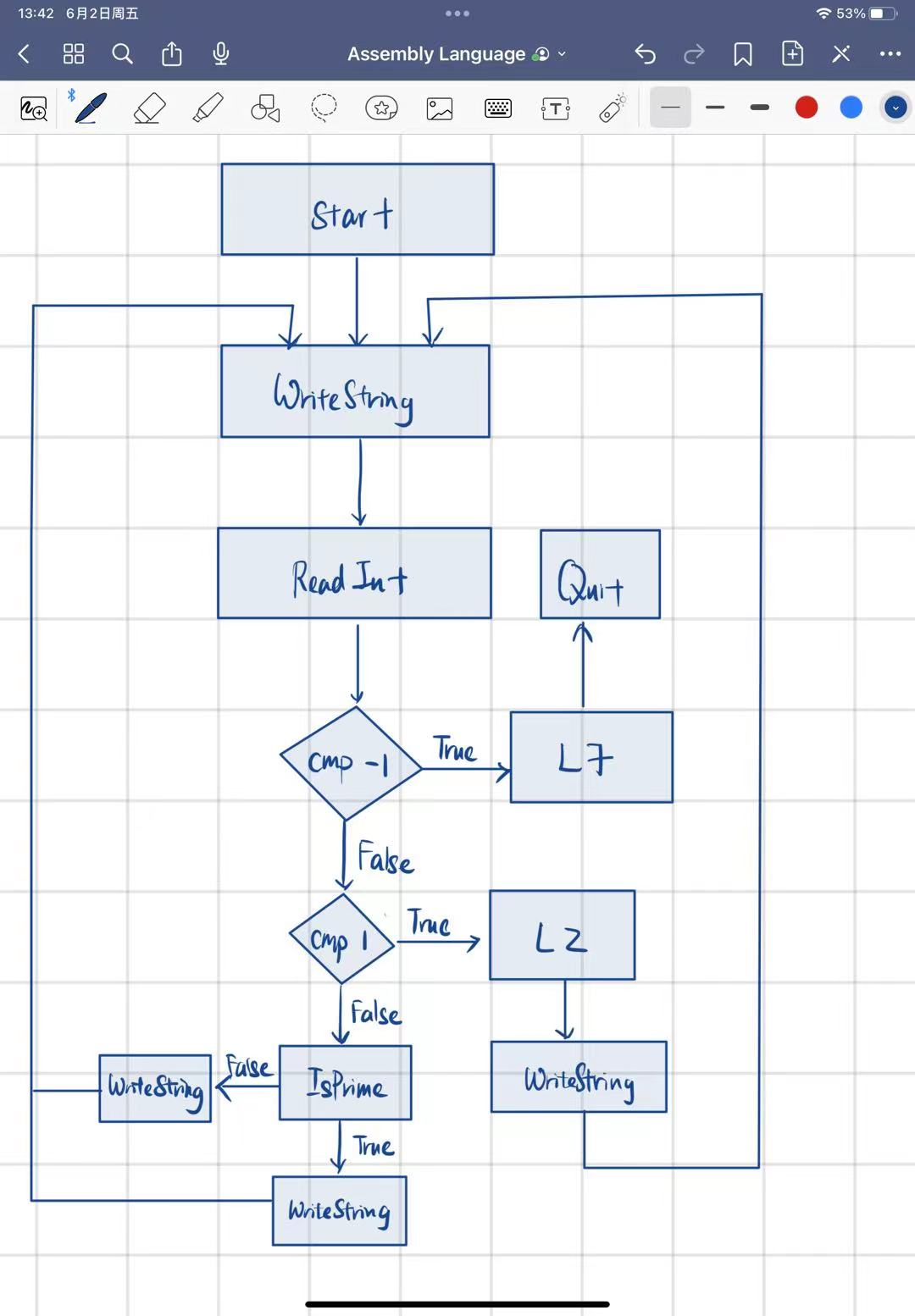
1. Software environment

Visual Studio 2008 and above versions of applications.

#### Experimental contents

1. **Prime number program**
2. Write a procedure named IsPrime that sets the Zero flag if the 32-bit integer passed in the EAX register is prime.
3. Write a test program that prompts the user for an integer, calls IsPrime, and displays a message indicating whether or not the value is prime. Continue prompting the user for integers and calling IsPrime until the user enters -1.

**Requirement: Draw the flowchart of your IsPrime procedure. Test your program using several inputs. Your report needs to include the flowchart of the IsPrime procedure, the source code of your program, and the input/output of your program in the consoler.**

****

include Irvine32.inc

.data

prompt BYTE "Enter a Integer:",0

prompt1 BYTE "Is not a prime",0

prompt2 BYTE "Is a prime",0

prompt3 BYTE "Invalid integer",0

.code

main PROC

L1:

mov edx,OFFSET prompt

call WriteString

call ReadInt

cmp eax,-1

je L7

cmp eax,1

je L2

call IsPrime

jmp L1

L2:

mov edx,OFFSET prompt3

call WriteString

call Crlf

jmp L1

L7:

ret

main endp

;---------------------------------------------------------

IsPrime PROC

; Set zero flag if integer is prime

; receives: integer

; returns: integer

;---------------------------------------------------------

cmp eax, 2 ; compare with 2

jb L6 ; if less then 2 then jump to L6

mov ecx, 2

L3:

xor edx, edx ; clear edx

div ecx ; divide

cmp edx, 0 ; check if remainder is zero or not

je L5 ; if is zero jump to L5 because it is not a prime number

inc ecx

cmp ecx, eax ; compare with the number

jbe L3 ; if ecx is less than or equal to eax then jump to L3

jmp L6 ; if have nonzero remainders thenjump to L6

L5:

mov edx, OFFSET prompt1

call WriteString

call Crlf

ret

L6:

mov edx, OFFSET prompt2

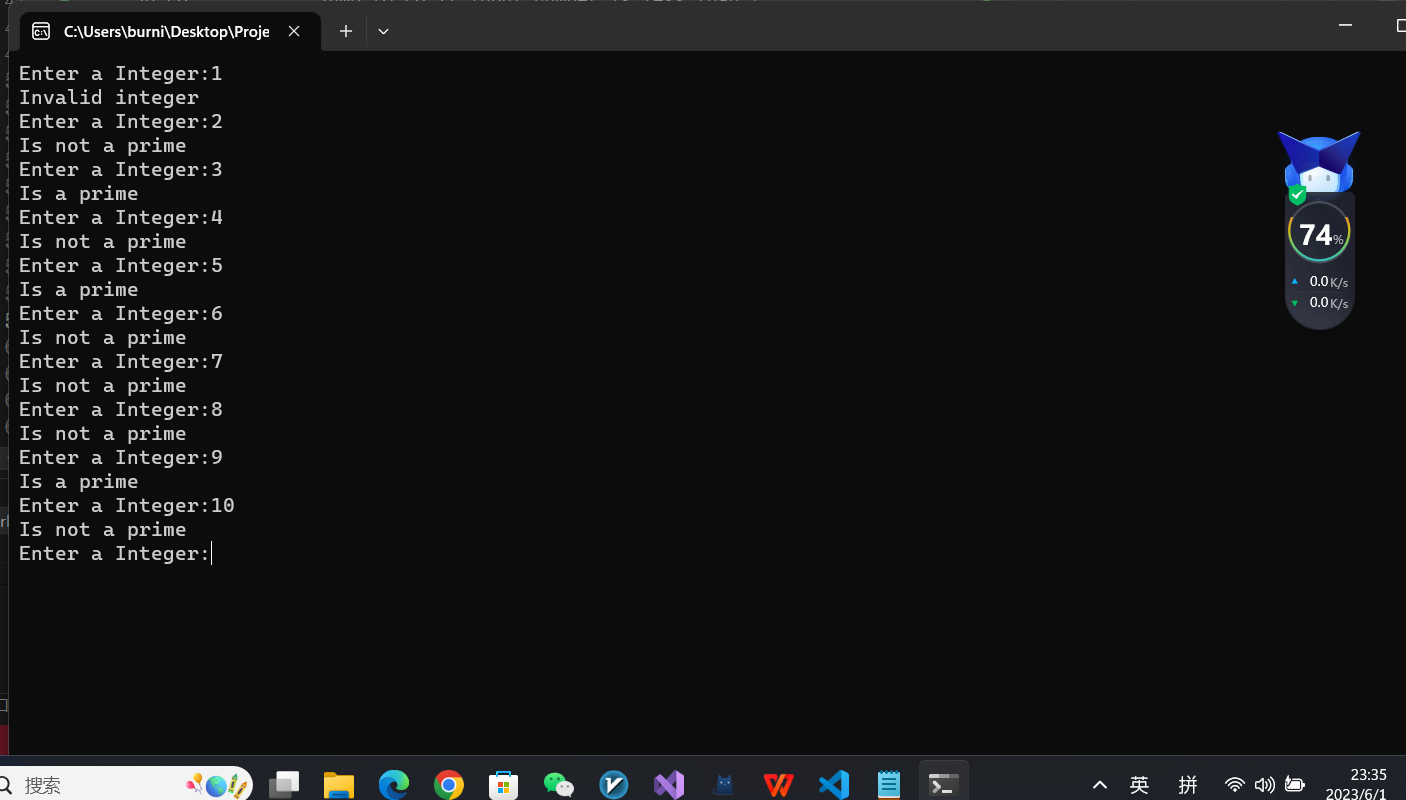
call WriteString

call Crlf

ret

IsPrime endp

end main



1. **Str\_remove Procedure**
2. Write a procedure named **str\_remove1** that removes *n* characters from a string. Pass a pointer to the position in the string where the characters are to be removed. Pass an integer specifying the number of characters to remove. The following code, for example, shows how to remove “xxxx” from **target**:

.data

target BYTE "abcxxxxdefghijklmop",0

.code

INVOKE str\_remove1, ADDR [target+3], 4

1. Write a procedure named **str\_remove2** that removes *n* characters from a string. Pass an integer specifying the position in the string from where the characters are to be removed. Pass an integer specifying the number of characters to remove. The following code, for example, shows how to remove “xxxx” from **target**:

.data

target BYTE "abcxxxxdefghijklmop",0

.code

INVOKE str\_remove2, ADDR target, 4, 4

1. Write a procedure named **main** that tests **str\_remove1** and **str\_remove2** using several different arguments and displays the strings before and after removing characters.

**Requirement: Your report needs to include the source code of your program and the output of your program in the consoler.**

include Irvine32.inc

.data

.code

main PROC

INVOKE str\_remove1,ADDR[target+3],4

INVOKE str\_remove2,ADDR target,4,4

main endp

;---------------------------------------------------------

str\_remove1 PROC

; remove n characters from a string

; receives: nothing

; returns: nothing

;---------------------------------------------------------

str\_remove1 endp

;---------------------------------------------------------

str\_remove2 PROC

; remove n characters from a string

; receives: nothing

; returns: nothing

;---------------------------------------------------------

str\_remove2 endp

end main