**Course: CEA201 – Computer Organization and Architecture**

**Item: Assembly assignments for FA22**

**Note:**

* Assignment #1 (vấn đề 1) và Assignment #2 (vấn đề 1) dành cho ½ nửa đầu danh sách các bạn sinh viên trong lớp

*Assignment #1 (issue 1) and Assignment #2 (issue 1) for the first half of the list of students in the class*

* Assignment #2 (vấn đề 2) và Assignment #2 (vấn đề 2) dành cho ½ nửa sau danh sách các bạn sinh viên trong lớp

*Assignment #2 (issue 2) and Assignment #2 (issue 2) for the second half of the list of students in the class*

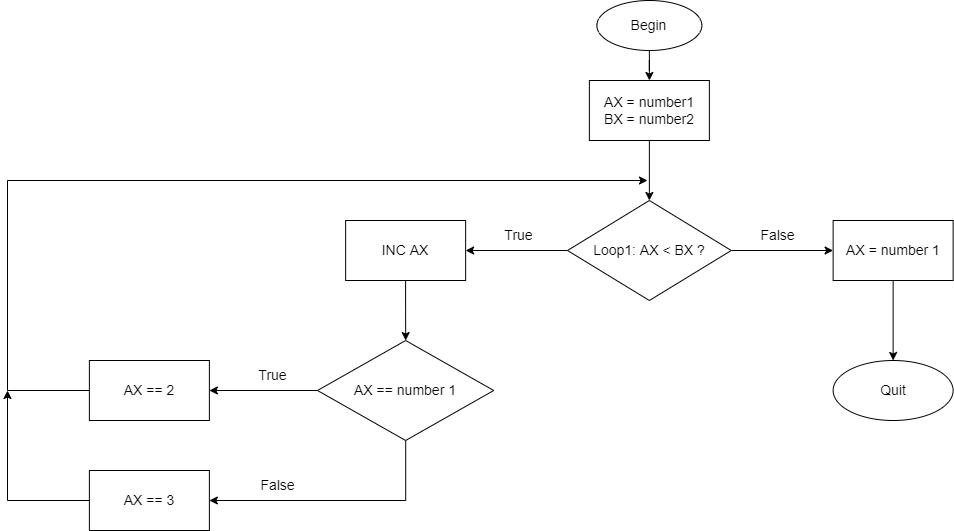
***Assignments #1***

Vấn đề 1: Viết chương trình sử dụng vòng lặp WHILE…DO hiển thị ký tự từ A 🡪 Z trong bảng mã ASCII

*Problem #1: Write a program using a WHILE…DO loop to print these characters from A to Z in ASCII table*

Vấn đề 2: Giả sử giải đồ thuật toán được cho như hình vẽ bên dưới. Viết chương trình thực hiện giải thuật này.

*Problem #2: Assume that, the flowchart of algorithm is given as shown below. Write a program to implement this algorithm*



***Assignments #2***

Vấn đề 1: Viết chương trình nhập vào một số hexa (<=2 chữ số), chuyển đổi số hexa vừa nhập thành số nhị phân và hiển thị số nhị phân vừa chuyển đổi ra màn hình

*Problem #1: Write a program to input a number by hexa (<=2 digit), convert this inputed number into binary format and print this number with its binary format on termnial screen*

Vấn đề 2: Giả sử màn hình termnial được hiển thị như sau:

*Problem #2: Assume that we have the terminal screen below:*

Enter number 1:

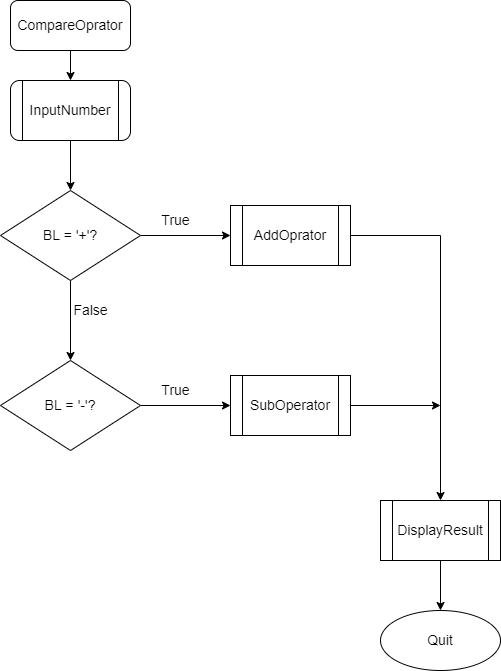
Enter number 2:

Enter your arithmetic operator (+, - ):

Sum:

Gợi ý chương trình như sau:

*Suggested the flowchart of program as follow:*



include \masm32\include\masm32rt.inc

.code

start:

call main

exit

main proc

LOCAL num1: DWORD

LOCAL num2: DWORD

LOCAL operator: DWORD

LOCAL sum: DWORD

LOCAL check: DWORD

mov num1, sval(input("Enter number 1: "))

mov num2, sval(input("Enter number 2: "))

mov operator, input("Enter your operator (+, -): ")

checkOP:

mov check, cmp$("+", operator)

cmp check, 0

je add2num

jne checksub

add2num:

mov eax, num1

mov ebx, num2

add eax, ebx

mov sum, eax

print chr$("Sum: ")

print str$(sum)

jmp stop

checksub:

mov check, cmp$("-", operator)

cmp check, 0

je sub2num

jne notexistOP

sub2num:

mov eax, num1

mov ebx, num2

sub eax, ebx

mov sum, eax

print chr$("Sum: ")

print str$(sum)

jmp stop

notexistOP:

print chr$("Operator is not supported!")

jmp stop

stop:

ret

main endp

end start