**Homework 5**

Author: Yifan Liu

Student ID: 518021910609

**Problem 1**. (Array initialization) Allocate 100 bytes of data, and assign value from 0-99 to those bytes.

**Solution.**

TITLE PROGRAM1-1 ARRAY-INITIALIZATION

PAGE 60,132

.MODEL SMALL

.STACK 64

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

.DATA

COUNT EQU 100

ARR DB 100 DUP(0)

NUM EQU 0

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

.CODE

MAIN PROC NEAR

MOV AX,SEG ARR

MOV DS,AX

MOV CX,COUNT

MOV BL,NUM

MOV SI,OFFSET ARR

BACK:MOV ARR[SI],BL

JNC OVER

INC AH

OVER:INC SI

INC BL

DEC CX

JNZ BACK

MOV AH,4CH

INT 21H

MAIN ENDP

END MAIN

**Problem 2**. (If/else translation) Allocate a word in the memory with any value and another word for storing its absolute value. Write the program to perform the conversion.

**Solution.**

TITLE PROGRAM1-2

PAGE 60,132

.MODEL SMALL

.STACK 64

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

.DATA

x DB 0;abs number

NUM DB 5;random number

data ends

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

.CODE

MAIN PROC FAR

MOV AX,WORD PTR data

MOV DS,AX

MOV AL,NUM

CMP AL,0

JGE LABE

NEG AL

LABE:MOV x,AL

MOV AH,4CH

INT 21H

MAIN ENDP

END MAIN

**Probem 3.** (Function argument and return value) Convert the following C code to the 8086 assembly code. You need to handle the function call argument passing and return value properly. [Hint: you can use stack for passing the argument and register AX for receiving the return value of the function.]

void main() {

int a, b, c, d;

a = -1;

b = 1;

c = abs(a);

d = add(a, b);

}

int abs(a) {

if (a > 0)

return a;

else

return -a;

}

int add(a, b) {

return a+b;

}

**Solution.**

TITLE 1-3

PAGE 60,132

.MODEL SMALL

.STACK 64

;----------

.DATA

a DB -1

b DB 1

c DB 0

d DB 0

.CODE

ABS PROC

MOV CL,AL

CMP CL,0

JGE LABE

NEG CL

LABE:RET

ABS ENDP

AD PROC

MOV CL,AL

ADD CL,BL

RET

AD ENDP

MAIN PROC FAR

MOV AX,SEG c

MOV DS,AX

MOV AL,a

MOV BL,b

CALL ABS

MOV c,CL

CALL AD

MOV d,CL

MOV AH,4CH

INT 21H

MAIN ENDP

END MAIN

**Problem 4.** (Function local variable) Convert the foo() function to assembly (no need to convert main). You need to handle the local variables properly. [Hint: you can use the stack for local variables and remember to clean the stack before return.]

void main() {

int a;

a = foo();

}

void foo() {

int a, b, c;

...

}

**Solution.**

a EQU WORD PTR [bp-2]

b EQU WORD PTR [bp-4]

c EQU WORD PTR [bp-6]

foo PROC

PUSH bp

MOV bp,sp

SUB sp,6

MOV a,1

MOV b,1

MOV c,1

MOV sp,bp

POP bp

RST

foo ENDP