School of Computing
Faculty of Engineering
UNIVERSITI TEKNOLOGI MALAYSIA

SUBJECT NAME: **COMPUTER ORGANIZATION AND ARCHITECTURE**

SUBJECT CODE: **SECR 2033**

SEMESTER: **2019/20-2**

LAB TITLE: **Programming 6: Conditional Structures**

STUDENT INFO :

Name	Matric No.
LOO ZHI XUEN	A19EC0078
KONG HAO YANG	A19EC0065
SEE WEN XIANG	A19EC0206

SECTION: **07**

SUBMISSION DATE: 23/06/2020

COMMENTS:

QUESTION 1

```
TITLE lab6 (main.asm)
INCLUDE Irvine32.inc
.data
MY_MONEY DWORD ?
YOUR_MONEY DWORD ?
YOU_DONATE DWORD ?
I_DONATE DWORD ?
instruction1 BYTE "Enter the value for MY_MONEY: ",0
instruction2 BYTE "Enter the value for YOUR_MONEY: ",0

.code
main PROC
mov edx,OFFSET instruction1
call writeString
call readDec
mov MY_MONEY,eax
mov edx,OFFSET instruction2
call writeString
call readDec
mov YOUR_MONEY,eax
cmp MY_MONEY,eax
jnb OUTT
mov YOU_DONATE,20
jmp DONE
OUTT:
mov YOU_DONATE,10
mov I_DONATE,10
DONE:
exit
main ENDP
END main
```

QUESTION 2

```
TITLE lab6 (main.asm)
INCLUDE Irvine32.inc
.data
instruction1 BYTE "Enter the value for EBX: ",0
instruction2 BYTE "Enter the value for ECX: ",0
instruction3 BYTE "Enter the value for EDX: ",0
.code
main PROC
mov edx,OFFSET instruction1
call writeString
call readDec
mov ebx,eax
mov edx,OFFSET instruction2
call writeString
call readDec
mov ecx,eax
mov edx,OFFSET instruction3
call writeString
call readDec
mov edx,eax
cmp ebx,ecx
jng OUTT
cmp ecx,edx
jnle OUTT
mov eax,9
mov eax,10
OUTT:


exit
main ENDP
END main
```

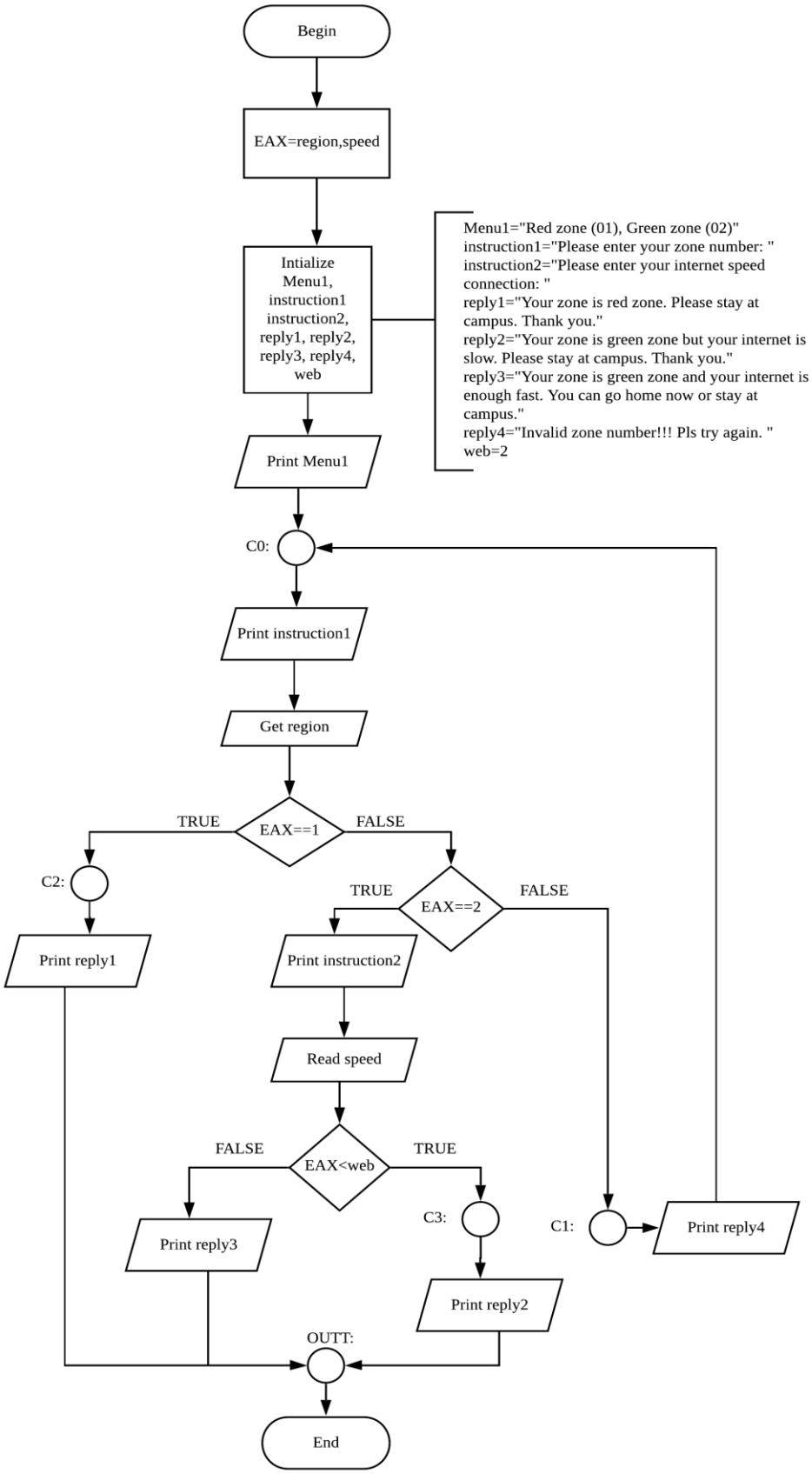
QUESTION 3

```
TITLE lab6 (main.asm)
INCLUDE Irvine32.inc

.data
MIN_CONTRIBUTION DWORD ?
MINE DWORD ?
instruction1 BYTE "Enter tha value for miminmum contribution: ",0
instruction2 BYTE "Enter tha value for my contribution: ",0
instruction3 BYTE "Sorry your contribution is not enough.",0
instruction4 BYTE "Thank you for your generous contribution.",0

.code
main PROC
mov edx,OFFSET instruction1
call writeString
call ReadDec
mov MIN_CONTRIBUTION,eax
mov edx,OFFSET instruction2
call writeString
call ReadDec
mov MINE,eax
C1:
	cmp MIN_CONTRIBUTION,eax
	jb C2
	mov edx,OFFSET instruction3
	call writeString
	call crlf
	inc eax
	jmp C1

C2:
	mov edx,OFFSET instruction4
	call writeString
	call crlf
exit
main ENDP
END main
```

QUESTION 4

```
TITLE lab6 (main.asm)
INCLUDE Irvine32.inc

.data
sum DWORD 0
sample DWORD 50
array DWORD 10,60,20,33,72,89,45,65,72,18
ArraySize = ($ - Array) / TYPE array

.code
main PROC
	mov eax,0
	mov edx,sample
	mov esi,0
	mov ecx,ArraySize
L1:
	cmp esi,ecx
	jae OUTT
	cmp array[esi*4],edx
	jbe L2
	add eax,array[esi*4]

L2: inc esi
	jmp L1

OUTT: mov sum,eax

exit
main ENDP
END main
```

QUESTION 5
A)


Begin
EAX=region,speed
Intialize Menu1, instruction1 instruction2, reply1, reply2, reply3, reply4, web
Menu1="Red zone (01), Green zone (02)"
instruction1="Please enter your zone number: "
instruction2="Please enter your internet speed connection: "
reply1="Your zone is red zone. Please stay at campus. Thank you."
reply2="Your zone is green zone but your internet is slow. Please stay at campus. Thank you."
reply3="Your zone is green zone and your internet is enough fast. You can go home now or stay at campus."
reply4="Invalid zone number!!! Pls try again. "
web=2
Print Menu1
C0:
Print instruction1
Get region
TRUE
EAX==1
FALSE
C2:
Print reply1
TRUE
EAX==2
FALSE
Print instruction2
Read speed
FALSE
EAX<web
TRUE
C3:
Print reply3
Print reply2
C1:
Print reply4
OUTT:
End

B)

```
TITLE lab2 (main.asm)

INCLUDE Irvine32.inc


.data

region DWORD ?

menu1 BYTE "Red zone (01), Green zone (02)",0

instruction1 BYTE "Please enter your zone number: ",0

instruction2 BYTE "Please enter your internet speed connection: ",0

reply1 BYTE "Your zone is red zone. Please stay at campus. Thank you.",0

reply2 BYTE "Your zone is green zone but your internet is slow. Please stay at campus.
Thank you.",0

reply3 BYTE "Your zone is green zone and your internet is enough fast. You can go home
now or stay at campus.",0

reply4 BYTE "Invalid zone number!!! Please try again. ",0

speed DWORD ?

web DWORD 2


.code

main PROC

mov edx,OFFSET menu1

call writeString

call CRLF

Call CRLF

C0:

mov edx,OFFSET instruction1

call writeString

call readInt

mov region,eax

cmp eax,1

je C2

cmp eax,2

jne C1

mov edx,OFFSET instruction2

call Writestring
```

```
call readInt
mov speed,eax
cmp eax,web
jb C3
mov edx,OFFSET reply3
call writeString
call crlf
jmp OUTT
C1: mov edx,OFFSET reply4
       call writeString
       call crlf
       jmp C0
C2: mov edx,OFFSET reply1
       call writeString
       call crlf
       jmp OUTT
C3: mov edx,OFFSET reply2
       call writeString
       call crlf
OUTT:

exit
main ENDP
END main
```