Manas Pratim Biswas

IT-A1-025

ASM ASSIGNMENT 03(KIT ASSIGNMENT)

---

1. Write the 8086 machine code to check whether a number is prime or not.

ASSEMBLY CODE	MACHINE CODE (2)	MACHINE CODE (16)
.model small	-	
.stack 100h	-	
.data	-	
.code	-	
main proc	-	
mov cx,19	10111001 00010011	B9H 13H 00H
mov bx,0	00000000	BBH 00H 00H
mov bl,2	10111011 00000000	B3H 02H
loop1:	00000000	
mov ax,cx	10110011 00000010	89H C8H
div bl	-	F6H F3H
cmp ah,0	10001001 11001000	80H FCH 00H
je isnotprime	11110110 11110011	74H 07H
inc bl	10000000 11111100	43H
cmp bx,cx	00000000	39H CBH
jge isprime	01110100 00000111	7DH 0BH
jmp loop1	01000011	EBH 90H
isnotprime:	00111001 11001011	
mov dx,'N'	01111101 00001011	BAH 4EH 00H
mov ah,2	11101011 10010000	B4H 02H
int 21h	-	CDH 21H
jmp endprog	10111010 01001110	EBH 07H
isprime:	00000000	
mov dx,'Y'	10110100 00000010	BAH 59H 00H
mov ah,2	11001101 00100001	B4H 02H
int 21h	11101011 00000111	CDH 21H
endprog:	-	
mov ah,4ch	10111010 01011001	B4H 4CH
int 21h	00000000	CDH 21H
main endp	10110100 00000010	

end main	11001101 00100001 - 10110100 01001100 11001101 00100001 - -	

2. Write the 8086 machine code to compute the difference of two numbers and store the magnitude and sign.

ASSEMBLY CODE	MACHINE CODE (2)	MACHINE CODE (16)
.model small	-	
.stack 100h	-	
.data	-	
.code	-	
main proc	-	
mov cl,3	10110001 00000011	B1H 03H
sub cl,5	10000000 11101001	80H E9H 05H
js label1	00000101	78H 08H
mov dl,'+'	01111000 00001000	B2H 2BH
mov ah,2	10110010 00101011	B4H 02H
int 21h	10110100 00000010	CDH 21H
jmp label2	11001101 00100001	EBH 08H
label1:	11101011 00001000	
mov dl,'-'	-	B2H 2DH
mov ah,2	10110010 00101101	B4H 02H
int 21h	10110100 00000010	CDH 21H
neg cl	11001101 00100001	F6H D9H
label2:	11110110 11011001	
mov dl,cl	-	89H CAH
add dl,48	10001001 11001010	80H C2H 30H
mov ah,2	10000000 11000010	B4H 02H
int 21h	00110000	CDH 21H
mov ah,4ch	10110100 00000010	B4H 4CH
int 21h	11001101 00100001	CDH 21H
main endp	10110100 01001100	
end main	11001101 00100001	
	-	

	-	

3. Write the 8086 machine code to count the number of "1"s present in a 16-bit number and store the counter in some memory location. (e.g. in 34(=100010), there are two "1"s)

ASSEMBLY CODE	MACHINE CODE (2)	MACHINE CODE (16)
.model small	-	
.stack 100h	-	
.data	-	
.code	-	
main proc	-	
mov ax,100010b	10111000 00100010	B8H 22H 00H
mov bl,10b	00000000	B3H 02H
mov cx,0	10110011 00000010	B9H 00H 00H
loop1:	10111001 00000000	
div bl	00000000	F6H F3H
cmp ah,1b	-	80H FCH 01H
jne jump1	11110110 11110011	75H 01H
inc cx	10000000 11111100	41H
jump1:	00000001	
mov ah,0	01110101 00000001	B4H 00H
cmp al,0	01000001	80H F8H 00H
je endprog	-	74H 02H
jmp loop1	10110100 00000000	EBH 91H
endprog:	10000000 11111000	
mov dx,cx	00000000	89H CAH
add dx,48	01110100 00000010	81H C2H 30H
	11101011 10010001	00H
mov ah,2	-	B4H 02H

int 21h	10001001 11001010	CDH 21H
mov ah,4ch	10000001 11000010	B4H 4CH
int 21h	00110000 00000000	CDH 21H
main endp	10110100 00000010	
end main	11001101 00100001	
	10110100 01001100	
	11001101 00100001	
	-	
	-	

4. Write the 8086 machine code to calculate the GCD of two unsigned 16-bit numbers.

ASSEMBLY CODE	MACHINE CODE (2)	MACHINE CODE (16)
.model small	-	
.stack 100h	-	
.data	-	
.code	-	
main proc	-	
mov bx,6	10111011 00000110	BBH 06H 00H
solve:	00000000	
mov ax,9	-	B8H 09H 00H
div bl	10111000 00001001	F6H F3H
cmp ah,0	00000000	80H FCH 00H
jne skip	11110110 11110011	75H 0AH
mov ax,6	10000000 11111100	B8H 06H 00H
div bl	00000000	F6H F3H
cmp ah,0	01110101 00001010	80H FCH 00H
je endsolve	10111000 00000110	74H 05H
skip:	00000000	
mov ah,0	11110110 11110011	B4H 00H
dec bl	10000000 11111100	4BH
jmp solve	00000000	EBH 99H
endsolve:	01110100 00000101	

mov dx,bx
add dx,48
 	
mov ah,2
int 21h
mov ah,4ch
int 21h
main endp
end main
	-
	10110100 00000000
	01001011
	11101011 10011001
	-
	10001001 11011010
	10000001 11000010
	00110000 00000000
	10110100 00000010
	11001101 00100001
	10110100 01001100
	11001101 00100001
	-
	-
		89H DAH
		81H C2H 30H
		00H
		B4H 02H
		CDH 21H
		B4H 4CH
		CDH 21H

5. Write the 8086 machine code to search the smallest number in an array of ten 8-bit numbers. The array of numbers (bytes) is stored in memory.

ASSEMBLY CODE	MACHINE CODE (2)	MACHINE CODE (16)

.model small
.stack 100h
.data
arr db
5,3,6,8,4,6,2,7,
9,8
.code
main proc
mov ax,@data
mov ds,ax
mov si,offset
arr
mov dh,0
mov dl,[si]
	-
	-
	-
	[stored in memory]
	
	-
	-
	-
	-
	10001001 00000110
	10110110 00000000
	10001010 00010100
	10111001 00001010
	00000000
		
		89H 06H
		B6H 00H
		8AH 14H
		B9H 0AH
		00H

mov cx,10	-	
loop1:	10001010 00011100	8AH 1CH
mov bl,[si]	00111000 11010011	38H D3H
cmp bl,dl	01111101 00000010	7DH 02H
jge skip1	10001000 11011010	88H DAH
mov dl,bl	-	
skip1:	01000110	46H
inc si	11100010 10001011	E2H 8BH
loop loop1	10000000 11000010	80H C2H
add dl,48	00110000	30H
mov ah,2	10110100 00000010	B4H 02H
int 21h	11001101 00100001	CDH 21H
mov ah,4ch	10110100 01001100	B4H 4CH
int 21h	11001101 00100001	CDH 21H
main endp	-	
end main	-	