

1. 다음 프로그램이 실행되는 과정에서 명령어 실행직후에 빈칸의 값들을 채우시오.  
 (8비트 연산임; 빼기는 2의 보수를 더하는 것으로 처리; 글자 ‘0’의 ASCII 코드는 0x30 )

	R2 (8비트)									CCR				
	이진수								정수	비부호 정수	N	Z	C	V
COPY R2, '2'														
SUB R2, R2, '0'														
ADD R0, R2, 127														
ASR R2, R0, 8													-	-
LSR R2, R0, 8													-	-
XOR R2, R2, -1														
AND R2, R2, 0x03														
LSL R2, R2, 2													-	-
OR R2, R2, 0x06														
BR p, 0x3000													-	-

2. 다음 프로그램에 대하여

(a) 명령어 실행 단계별로 메모리 동작에 대하여 빈칸을 채우시오. (해당사항이 없으면 ‘없음’, 기계어 코드이면 ‘기계어’로 표기; 주소와 데이터는 메모리 읽기나 쓰기 동작 시에 메모리에 전달되는 주소와 읽기/쓰기의 내용을 의미함)

프로그램	실행단계	메모리 동작 (읽기/쓰기/없음)	주소	데이터
.ORIGIN 4000	-	-	-	-
COPY R0, 5	명령어 읽기			
	명령어 실행			
STORE R0, y	명령어 읽기			
	명령어 실행			
LEA R1, y	명령어 읽기			
	명령어 실행			
LEA R2, x	명령어 읽기			
	명령어 실행			
STR R1, R2, 0	명령어 읽기			
	명령어 실행			
LDR R1, R2, 2	명령어 읽기			
	명령어 실행			
STR R1, R2, 1	명령어 읽기			
	명령어 실행			
x: .BLOCK 2 y: .BLOCK 1	-	-	-	-

(b) 마지막 명령어를 실행한 시점에 x와 y부분의 값을 모두 나열하시오.

3. 다음의 프로그램을 완성하시오. (프로그램은 메모리의 2000 번지부터 적재)

프로그램 적재 주소 지정; 글자 입력 받아서 변수 input에 저장	
글자가 'A' 보다 작으면 프로그램 종료	
글자의 최하위 3비트만 매스킹하여 (정수 0~7이 됨) 변수 ch1에 저장	
글자를 3비트 오른쪽으로 쉬프트하여 최하위 3비트만 매스킹하여 변수 ch2에 저장	
글자 '0' 을 출력 변수 ch2 값을 글자로 변환하여 출력 변수 ch1 값을 글자로 변환하여 출력	
	SWI 255
필요한 변수들 모두 선언	

<참고자료>

어셈블리어 표현	의미
ADD Rd, Rs, Rs2	$Rd \leftarrow Rs + Rs2$
SUB Rd, Rs, Rs2	$Rd \leftarrow Rs - Rs2$
CMP Rs, Rs2	$Rs - Rs2$
AND Rd, Rs, Rs2	$Rd \leftarrow Rs \text{ AND } Rs2$
OR Rd, Rs, Rs2	$Rd \leftarrow Rs \text{ OR } Rs2$
XOR Rd, Rs, Rs2	$Rd \leftarrow Rs \text{ XOR } Rs2$
NOT Rd, Rs	$Rd \leftarrow \text{NOT } Rs$
LSL Rd, Rs, n	$Rd \leftarrow Rs$ 를 n비트 왼쪽이동
LSR Rd, Rs, n	$Rd \leftarrow Rs$ 를 n비트 오른쪽이동
ASL Rd, Rs, n	$Rd \leftarrow Rs$ 를 n비트 왼쪽이동
ASR Rd, Rs, n	$Rd \leftarrow Rs$ 를 n비트 오른쪽이동 (부호유지)
COPY Rd, Rs2	$Rd \leftarrow Rs2$
LOAD Rd, addr	$Rd \leftarrow$ 메모리의 addr 번지 내용
LDR Rd, Ra, offset	$Rd \leftarrow$ 메모리의 $Ra + \text{offset}$ 번지 내용
STORE Rs, addr	메모리의 addr 번지 내용 $\leftarrow Rs$
STR Rs, Ra, offset	메모리의 $Ra + \text{offset}$ 번지 내용 $\leftarrow Rs$
BR nzp, addr	조건에 따라 실행 위치를 addr 주소로 이동
LEA Rd, addr	$Rd \leftarrow$ addr 번지
SWI 0	키보드에서 한 문자를 읽어서 R0에 기록, 입력한 문자는 화면에도 출력
SWI 1	R0에 저장된 문자를 화면에 출력
SWI 255	실행 중인 프로그램의 종료

4. 다음에 대해 간단히 설명하시오.

분기 명령어의 용도	
직접 모드와 레지스터 간접모드 차이	
LOAD와 LEA의 차이	
NZCV 비트별 설명	
지시어	