3. Subroutine question: A programmer wrote a small subroutine to wait for a character in the UART. This routine is called "routin" in the code below. The programmer then called this routine in his larger character handling routine. This code fragment is as follows:

```
Address  BitPattern          Instruction
                             .set LSR,0x1014
                             .set RBR,0x1000
ffff0120 60000000            nop
ffff0124 48000065            bl routin
ffff0128 60000000            nop


ffff0188 3D8083E0  routin:   lis r12,0x83e00000@h
ffff018c 618C0000            ori r12,r12,0x83e00000@l
ffff0190 812C1014  again:    lwz r9,LSR(r12)
ffff0194 71290001            andi. r9,r9,0x01
ffff0198 4082FFF8            bne 0,again
ffff019c 80EC1000            lwz r7,RBR(r12)
ffff01a0 4E800020            blr
```

This question deals with register useage in the routine. Below is a before and after representation for 16 registers, half of the general purpose registers available. The before values are given (values of registers before executing the instruction at 0xffff0124). Your task is to fill in the after values (values of registers after executing the instruction at 0xffff0124 and beginning to execute the instruction located at 0xffff0128). Only mark in the After area those registers that have changed.

Additional information that has to do with your answer: the UART Line Status Register is available at offset 0x1014; the Receiver Buffer Register of the UART is available at offset 0x1000; the base address of the UART is 0x83e00000.

Before		After	
r0 = 0x00000000	r1 = 0x11111111	r0 =	r1 =
r2 = 0x22222222	r3 = 0x33333333	r2 =	r3 =
r4 = 0x44444444	r5 = 0x55555555	r4 =	r5 =
r6 = 0x66666666	r7 = 0x77777777	r6 =	r7 =
r8 = 0x88888888	r9 = 0x99999999	r8 =	r9 =
r10 = 0xAAAAAAAA	r11 = 0xBBBBBBBB	r10 =	r11 =
r12 = 0xCCCCCCCC	r13 = 0xDDDDDDDD	r12 =	r13 =
r14 = 0xEEEEEEEE	r15 = 0xFFFFFFFF	r14 =	r15 =
LR = 0x00000000	CTR = 0x00000000	LR =	CTR =

see back