Question 2

Code Explanation

To accomplish the program, we divide it into several tasks:

1. Load initial pattern I and shift amount k
2. Build a function f for single-bit right shift
3. Repeat f on target pattern T for k times
4. Store result T to memory

Among them, step 2 is the bottleneck. Further details are as follow:

1. Construct a total pattern S whose value is x0000
2. Construct a mask pattern M whose value is x0001
3. Test current bit by AND T with left shifted M
4. If result is non-zero, current bit is valid and accumulate M to S
5. Left shift M by one bit and repeat operations above for 15 times
6. Store S back to T and quit this function

With counters and labels added, this program can work properly.

Appendix: Assembly Code

```
        .ORIG   x3000
        LDI     R0,INP      ; R0 is target pattern
        LDI     R1,AMT      ; R1 is shift counter
LOOP0   BRNZP   LSHIFT      ; Do single right shift
CONT0   ADD     R1,R1,#-1   ; Decrease shift counter
        BRP     LOOP0       ; Repeat certain times
        STI     R0,RES      ; Store result to memory
        HALT                ; Terminate main part
LSHIFT  AND     R2,R2,#0    ; R2 is total pattern
        AND     R3,R3,#0
        ADD     R3,R3,#1    ; R3 is mask pattern
        AND     R4,R4,#0
        ADD     R4,R4,#15   ; R4 is bit counter
LOOP1   ADD     R5,R3,R3    ; R5 is one-bit higher
        AND     R5,R0,R5
        BRZ     CONT1       ; Test higher bit
        ADD     R2,R2,R3    ; Accumulate valid bit
CONT1   ADD     R3,R3,R3    ; Left shift mask
        ADD     R4,R4,#-1   ; Decrease bit counter
        BRP     LOOP1       ; Repeat certain times
        ADD     R0,R2,#0    ; Store back to R0
        BRNZP   CONT0       ; Go back to main part
INP     .FILL   x3100       ; x3100 initial pattern
AMT     .FILL   x3101       ; x3101 shift amount
RES     .FILL   x3102       ; x3102 result storage
        .END
```