How to build the mask dynamically? One possible approach shown below.		
This is not a complete code. Other solutions are possible. Sample registers are used.		
accepting input data:		
enter p: 173	0000 0000 0000 0000 0000 0000 1010 1101	p goes in $a2
enter n: 5	0000 0000 0000 0000 0000 0000 0000 0101	n goes in $a0
enter m: 3	0000 0000 0000 0000 0000 0000 0000 0011	m goes in $a1
how to build the mask?	1111 1111 1111 1111 1111 1111 1111 1111	fill $t0 with ones, we use it to build the mask
	bit pattern not important here	$t1 gets 32-n (in example: 32-5=27)
	0000 0000 0000 0000 0000 0000 0001 **1111**	shift $t0 right by $t1 (=27) **mask is ready!**
	0000 0000 0000 0000 0000 0000 **1111** 1000	shift $t0 left by $a1 (=3), **mask is in place!**
		you have to use variable (not immediate) shifts
how to use the mask?	0000 0000 0000 0000 0000 0000 1010 1101	value of p is in $a2 (in this example p=173)
	0000 0000 0000 0000 0000 0000 **1010** 1000	$t2 gets logical AND of p and mask
answers:		
"10101" - what unsigned integer is it?	0000 0000 0000 0000 0000 0000 0001 **0101**	shift $t2 right by $a1 (=3)
"10101" - what signed integer is it?	**1010** 1000 0000 0000 0000 0000 0000 0000	shift left logical (how far?)
	1111 1111 1111 1111 1111 1111 1111 **0101**	shift right arithmetic (how far?)