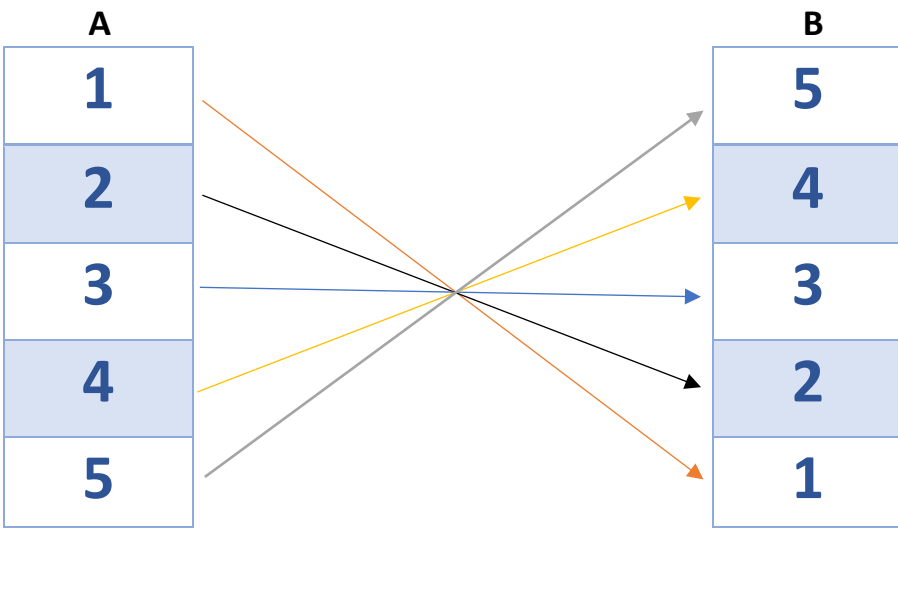


LIST OF PROGRAMS

Reversing an array program

	<pre>addi \$t0 \$s0 0 addi \$t1 \$zero 4 sll \$t1 \$t1 2 add \$t1 \$t1 \$s1 #loop (-6 from beq): lw \$t2 0(\$t0) sw \$t2 0(\$t1) addi \$t0 \$t0 4 addi \$t1 \$t1 -4 slt \$t3 \$t1 \$s1 beq \$t3 \$zero -6</pre>
<p>A is a 5-integer array whose starting address is in \$s0, the values 1,5 are stored in A in ascending order.</p> <p>Assume that B is another 5-integer array whose starting address is in \$s1</p> <p>\$s0 holds the base address which is 1000 \$s1 holds the base address which is 2000</p> <p>First, we will load the values 1,2,3,4,5 in the memory respectively starting from the base address 1000 which is stored in \$s0, and then we will load the base address 2000 into \$s1, before starting writing the program</p> <p>the values stored in array A will be stored in reverse order in array B</p>	

Program to test the rest of instructions

500	ori \$t0 \$zero 7	# \$t0 = 7
504	or \$t1 \$t0 \$t1	# \$t1 = 7
508	jal 130	# 520/4 = 130
512	nor \$t4 \$t2 \$t1	# \$t4 = -8
516	j 133	# 532/4 = 133
520	andi \$t2 \$t0 6	# \$t2 = 6
524	and \$t3 \$t2 \$t1	# \$t3 = 6
528	jr \$ra	# \$ra = 512
532	addi \$t5 \$t0 10	# \$t5 = 17
536	#end	# pc+4