```
.data
  A: .word 1, -2, 3, -4, 7
  Result: .asciiz "\n The Positve Sum is: "
.text
main: la $a0, A
  li $a1, 5
  li $t5, 0
  j print
  cont:
  la $a0, A
  jal fun
  j end

print: move $t0, $a0
print2:
lw $t2, 0($t0)
move $a0, $t2
li $v0, 1
syscall
li $a0, ' '
li $v0, 11
syscall
addi $t5, $t5, 1
addi $t0, $t0, 4
blt $t5, $a1, print2
j cont


fun: li $t0, 0 #sum
  li $t1, 0 #counter
next: lw $t2, 0($a0) #get int
  bltz $t2, Neg
  add $t0, $t0, $t2
  skip:
  addi $a0, $a0, 4
  addi $t1, $t1, 1
  blt $t1, $a1, next
  move $v0, $t0
  jr $ra

Neg:
  j skip


end:
  move $t0, $v0
  la $a0, Result
  li $v0, 4
  syscall
  move $a0, $t0
  li $v0, 1
  syscall
  li $v0, 10
  syscall
```


Console
1 -2 3 -4 7
The Positve Sum is: 11

For the single-cycle MIPS processor, what changes would be required of the

datapath in order to execute the followng instructions?

(a) blez ....

(b) jr ...

In each case, explain why the changes are necessary and how the instruction

would be executed in the modified datapath.

a) The data path to handle blez: blez handles one register. It branches if the register is less than or equal to zero. When such is the case, it branches by adding the branch offset to the program counter (PC). The offset is the number of instructions to branch past and is signed. So if $s <= 0, PC' = PC + 4 + (4 * offset). Else, PC' = PC + 4.
b) The data path to handle jr: The jump register simply jumps to the address contained in $s0. As such, PC = 'PC and 'PC = $s.