```
#Agha Noor Ahmed Khan

.data

   list1: .word 54,26,93,17,77,31,44,55,20
   length1: .word 9
   list2: .word 9,8,7,6,5
   length2: .word 5

   newline: .asciiz "\n"

.text

###############BEGIN main###############
main:
   la $a0, list1
   lw $a1, length1
   jal double_items

   la $a0, list2
   lw $a1, length2
   jal double_items

   la $a0, list1
   lw $a1, length1
   jal print_list

   li $v0, 4 #code for printing a string
   la $a0, newline
   syscall

   la $a0, list2
   lw $a1, length2
   jal print_list

   j exit_program
###############END main###############


###############BEGIN double_value###############
double_value:
   add $v0, $a0, $a0
   jr $ra
###############END double_value###############


###############BEGIN double_items###############
double_items:
   # Save $ra and $s registers on the stack
   addi $sp, $sp, -8       #Make space on the stack
   sw $ra, 4($sp)          #Save return address
   sw $s0, 0($sp)          #Save $s0

   move $s0, $a0           #Store the base address of the list in $s0
   move $s1, $a1           #Store the list length in $s1
   li $t0, 0               #Initialize the index counter i

double_items_loop:
   bge $t0, $s1, double_items_end   #Exit loop if i >= list length

   sll $t1, $t0, 2         #Multiply i by 4 to get the memory offset
   add $t2, $s0, $t1       #Calculate address of a_list[i]

   lw $a0, 0($t2)          #Load a_list[i] into $a0
   jal double_value        #Call double_value(a_list[i])

   sw $v0, 0($t2)          #Store the doubled value back in a_list[i]

   addi $t0, $t0, 1        #Increment the index counter
   b double_items_loop     #Repeat the loop

double_items_end:
   # Restore $ra and $s0 from the stack
   lw $ra, 4($sp)
   lw $s0, 0($sp)
   addi $sp, $sp, 8        #Restore stack pointer

   jr $ra                  #Return from the function
###############END double_items###############


###############BEGIN print_list###############
print_list:
   move $t0, $a0 #$a0 is the list address
   move $t1, $a1 #$a1 is the list length
   li $t2, 0 #loop counter i
```

```
print_list_loop:
   bge $t2, $t1, print_list_end #exit loop if i >= list length

   sll $t4, $t2, 2 #multiply loop counter by 4 to get memory offset
   add $t3, $t0, $t4 #add base address of list and memory offset

   lw $a0, 0($t3) #load a_list[i]
   li $v0, 1 #code for printing an integer
   syscall

   li $v0, 4 #code for printing a string
   la $a0, newline
   syscall

   addi $t2, $t2, 1 #increment loop counter

   b print_list_loop

print_list_end:
   jr $ra
###############END print_list###############

exit_program:
```