```
###################################
#void swap_genuine(int *x,int *y){
# int temp;
# temp=*x;
# *x=*y;
# *y=temp;
# return;
#}

#void main(){
# x=10;
# y=20;
# swap_genuine(&x,&y);
#}


## Program to swap two memory words
 .data # load data

 .text
 .globl main
main:
 #addi $sp,$sp,8 # allocate memory for local variables x and y
 addi $s0,$zero,10 # x -> $s0
 addi $s1,$zero,20  # y -> $s1
 sw   $s0,0($sp)
 sw   $s1,4($sp)
 add  $a0,$sp,$zero
 addi  $a1,$sp,4

 #addi $sp,$sp,8  # allocate memory for passing x and y parameters ($a0,$a1))
 sw   $a0,8($sp)
 sw   $a1,12($sp)
 jal  swap_genuine
 #addi $sp,$sp,-8  # pop of the two arguments

 li $v0, 10   # Sets $v0 to "10" to select exit syscall
 syscall # Exit


swap_genuine:
 #add  $sp,$sp,4   # allocate memory for return address
 sw   $ra,16($sp)
 #add  $sp,$sp,8  # allocate memory for saved registers $s3,$s4
 sw   $s3, 20($sp)
 sw   $s4, 24($sp)
 #add  $sp,$sp,4  # allocate memory for local variable temp

 lw  $s3,8($sp)  # temp = *x
 lw  $s3,0($s3)
 sw  $s3,28($sp)

 lw  $s4,12($sp)  # *x = *y
 lw  $s4,0($s4)
 lw  $t1,8($sp)
```

```
sw  $s4,0($t1)


lw  $s4,12($sp)
sw  $s3,0($s4)  # *y = temp

lw  $ra,16($sp)
#addi $sp,$sp, -16
jr  $ra
```