

Clayton, CS 2318-004, Assignment 2 Part 1C

This program takes an array of intergers, swaps around values and prints out the reversed and swapped array

```
.data
intArr:      .word 111, 222, 333, 444
seperator:   .asciiz ", "
arrOutput:   .asciiz "Here are the array contents: "
arrOutputRev: .asciiz "Here are the array contents swapped & reversed: "
swapMsg:     .asciiz "swapping... \n"

.text
.globl main

main:

#print arr

    li $v0, 4
    la $a0, arrOutput
    syscall #initial label

    li $v0, 1
    la $a0, intArr
    lw $t0, 0($a0)
    move $a0, $t0
    syscall # arr[0]
    li $v0, 4
    la $a0, seperator
    syscall

    li $v0, 1
    la $a0, intArr
    lw $t0, 4($a0)
    move $a0, $t0
    syscall # arr[1]
    li $v0, 4
    la $a0, seperator
    syscall

    li $v0, 1
    la $a0, intArr
    lw $t0, 8($a0)
    move $a0, $t0
    syscall # arr[2]
    li $v0, 4
    la $a0, seperator
    syscall

    li $v0, 1
    la $a0, intArr
    lw $t0, 12($a0)
```

```
move $a0, $t0
syscall # arr[3]
li $v0, 11
li $a0, '\n'
syscall
```

```
#swap arr
```

```
li $v0, 4
la $a0, swapMsg
syscall
```

```
la $a0, intArr
```

```
lw $t0, 0($a0)
lw $t1, 8($a0)
```

```
sw $t0, 8($a0)
sw $t1, 0($a0)
```

```
lw $t0, 4($a0)
lw $t1, 12($a0)
```

```
sw $t0, 12($a0)
sw $t1, 4($a0)
```

```
#print arr swapped and reversed
```

```
li $v0, 4
la $a0, arrOutputRev
syscall #initial label
```

```
li $v0, 1
la $a0, intArr
lw $t0, 12($a0)
move $a0, $t0
syscall # arr[0]
li $v0, 4
la $a0, seperator
syscall
```

```
li $v0, 1
la $a0, intArr
lw $t0, 8($a0)
move $a0, $t0
syscall # arr[1]
li $v0, 4
la $a0, seperator
syscall
```

```
li $v0, 1
```

```
la $a0, intArr
lw $t0, 4($a0)
move $a0, $t0
syscall # arr[2]
li $v0, 4
la $a0, seperator
syscall
```

```
li $v0, 1
la $a0, intArr
lw $t0, 0($a0)
move $a0, $t0
syscall # arr[3]
```

```
li $v0, 10
syscall
```

Output:

Here are the array contents: 111, 222, 333, 444

swapping...

Here are the array contents swapped & reversed: 222, 111, 444, 333

-- program is finished running --