```
.data
EmpArr: .space 480
newline:  .asciiz"\n"
name:   .asciiz "\nName: "
age:      .asciiz "Age: "
salary: .asciiz "Salary: "
ageN:    .asciiz "\nAge: "
salaryN: .asciiz "\nSalary: "
storeRec:.asciiz "Storing Record:\n"
printRec:.asciiz "\nPrinting Records:\n"
swapRec: .asciiz "\nAdjacent Record Swapping:\n"
swpPrompt:.asciiz "\nRecord number to be swapped.Valid Records are 0 to 9: "
outBounds: .asciiz "\nRecord out bounds."
notAdj: .asciiz "\nRecords are not adjacent."
                                .globl main
.text
main:
        la $t0, EmpArr
        li $t1,10
        #print title for part 1
        li $v0, 4
        la $a0, storeRec
        syscall
lLoad:
        blez $t1, part2
        #print name prompt
        li $v0, 4
        la $a0, name
        syscall
        #read and store the name of the record.
        move $a0, $t0
        li $a1, 40
        li  $v0, 8
        syscall
        #print age prompt
        li $v0, 4
        la $a0, age
        syscall
        #reads and store the age of the record.
        li $v0, 5
        syscall
        sw $v0, 40($t0)
        #print salary prompt
        li $v0, 4
```

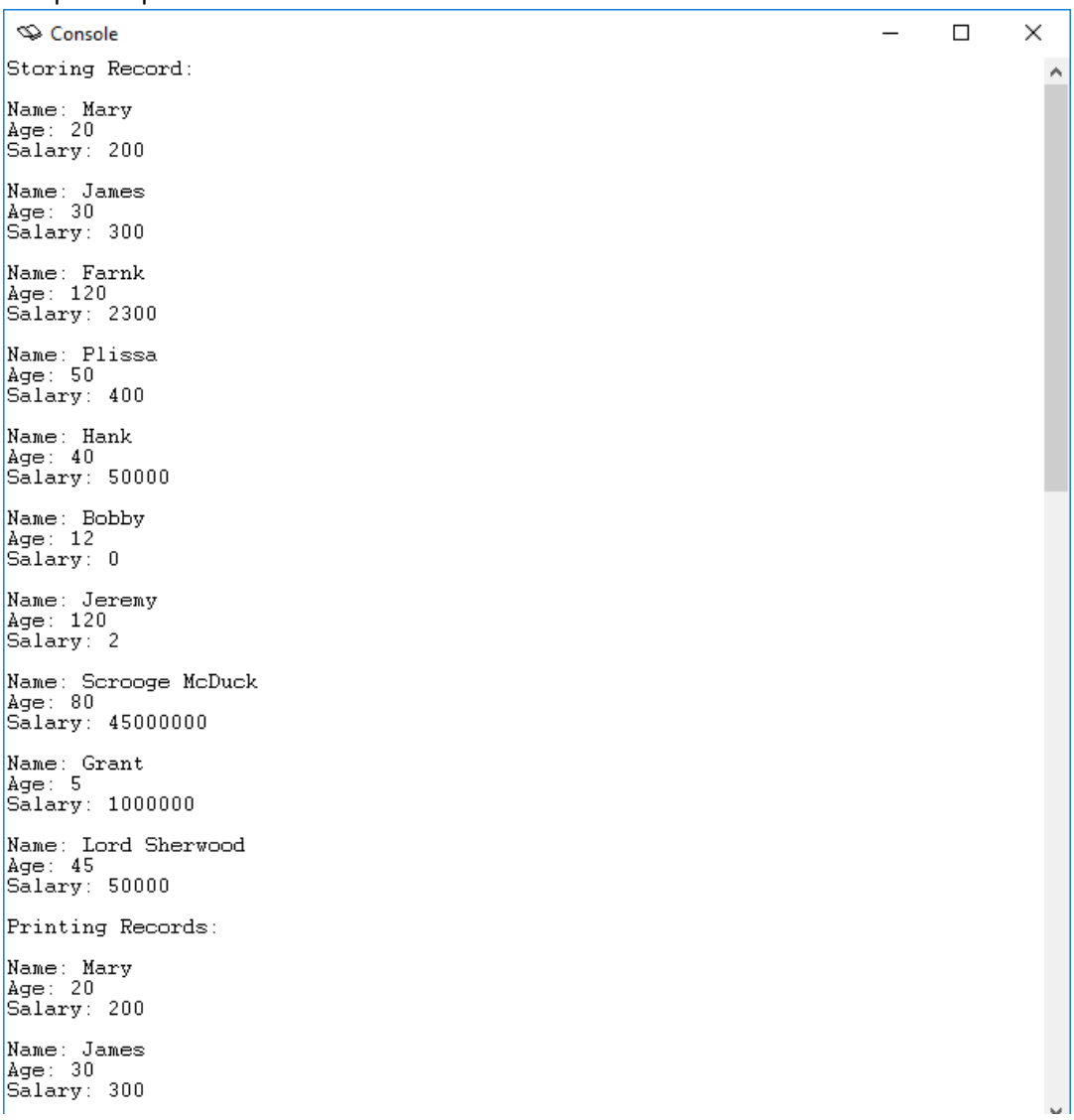
```
        la $a0, salary
        syscall
        #read and store the salary of the second record.
        li $v0, 5
        syscall
        sw $v0, 44($t0)
        #increment
        addi $t0,$t0,48
        addi $t1,$t1,-1
        b lLoad

part2:
        #print title for part 2
        li $v0, 4
        la $a0, printRec
        syscall
        la $a0, EmpArr
        li $a1,10
        jal print

part3:
        #ask for records to be swapped
        #print swap prompt
        li $v0, 4
        la $a0, swpPrompt
        syscall
        #reads and store the number of the record.
        li $v0, 5
        syscall
        move $t0,$v0
        #print swap prompt
        li $v0, 4
        la $a0, swpPrompt
        syscall
        #reads and store the number of the record.
        li $v0, 5
        syscall
        move $t1,$v0

        la $a0, EmpArr
        li $a1,10
        move $a2,$t0
        move $a3,$t1
        jal swapAdj
```

```
        li $v0,10
        syscall #exit
print:
        move $t0,$a0
        move $t2,$a1
lPrint:
        blez $t2,doneP
        # print the name of record.
        li $v0,4
        la $a0,name
        syscall
        li $v0,4
        move $a0, $t0
        syscall
        #print the age of the record.
        li $v0, 4
        la $a0, age
        syscall
        li $v0, 1
        lw $t1, 40($t0)
        move $a0, $t1
        syscall
        # print the salary of the record.
        li $v0, 4
        la $a0, salaryN
        syscall
        li $v0, 1
        lw $t1, 44($t0)
        move $a0, $t1
        syscall
        # start a new line
        li $v0,4
        la $a0, newline
        syscall
        #increment
        addi $t0,$t0,48
        addi $t2,$t2,-1
        b lPrint
doneP:
        jr $ra
swapAdj:
        #a0-array,a1-size of array
        #a2-index1,a3-index2
        move $t0,$a0
```

```
        move $t1,$a1
        #print title for part
        li $v0, 4
        la $a0, swapRec
        syscall
        #check if records are in bounds ex:(0...9)
        bltz $a2,oB
        bltz $a3,oB
        bge $a2,$t1,oB
        bge $a3,$t1,oB
        b checkAdj
oB:     li $v0, 4 #print out of bounds error and quit
        la $a0, outBounds
        syscall
        b ret
checkAdj:#check if records are adjacent
        subu $t2,$a2,$a3
        abs $t2,$t2 #gets the absolute difference
        beq $t2,1,isAdjacent
        li $v0, 4 #if not adjacent print error and quit
        la $a0, notAdj
        syscall
        b ret
isAdjacent:
        #get positions of entries in the array
        mul $t2,$a2,48
        mul $t3,$a3,48
        add $t2,$t2,$t0
        add $t3,$t3,$t0
        #create temporary copy of record in index1
        lw $t4,40($t2)
        lw $t5,44($t2)
        #copy record2 to record1
        lw $t7, 40($t3)
        sw $t7, 40($t2)
        lw $t6, 44($t3)
        sw $t6, 44($t2)
        #copy temp copy to record2
        sw $t4, 40($t3)
        sw $t5, 44($t3)
        #begin to swap names
        li $t6,39
swapChars:
        bltz $t6,printArray
```

```
        lb $t4,0($t2)
        lb $t5,0($t3)
        sb $t4,0($t3)
        sb $t5,0($t2)
        addi $t2,$t2,1
        addi $t3,$t3,1
        addi $t6,$t6,-1
        b swapChars
printArray:
        move $a0,$t0    #print entire array
        move $a1,$t1
        jal print
ret:jr $ra #exit
```

Sample Output:



```
Name: Farnk
Age: 120
Salary: 2300

Name: Plissa
Age: 50
Salary: 400

Name: Hank
Age: 40
Salary: 50000

Name: Bobby
Age: 12
Salary: 0

Name: Jeremy
Age: 120
Salary: 2

Name: Scrooge McDuck
Age: 80
Salary: 45000000

Name: Grant
Age: 5
Salary: 1000000

Name: Lord Sherwood
Age: 45
Salary: 50000
Record number to be swapped.Valid Records are 0 to 9: 7

Record number to be swapped.Valid Records are 0 to 9: 8

Adjacent Record Swapping:

Name: Mary
Age: 20
Salary: 200

Name: James
Age: 30
Salary: 300

Name: Farnk
Age: 120
Salary: 2300

Name: Plissa
Age: 50
Salary: 400

Name: Hank
Age: 40
Salary: 50000

Name: Bobby
Age: 12
Salary: 0

Name: Jeremy
Age: 120
Salary: 2

Name: Grant
Age: 5
Salary: 1000000

Name: Scrooge McDuck
Age: 80
Salary: 45000000

Name: Lord Sherwood
Age: 45
Salary: 50000
```

```
Record number to be swapped.Valid Records are 0 to 9: 0
Record number to be swapped.Valid Records are 0 to 9: 10
Adjacent Record Swapping:
Record out bounds.
```

```
Record number to be swapped.Valid Records are 0 to 9: -5
Record number to be swapped.Valid Records are 0 to 9: 5
Adjacent Record Swapping:
Record out bounds.
```

```
Record number to be swapped.Valid Records are 0 to 9: 1
Record number to be swapped.Valid Records are 0 to 9: 9
Adjacent Record Swapping:
Records are not adjacent.
```