# Author : Kevin De La Torre

# Class : CS 264

# Professor: Salloum

# Project : Lab 5

.data

# Records consist of 1 String ( 40 char max ), and 2 ints

recordData: .space 480 # 10 records, 48 bytes per record

tmpRecord: .space 48 # Used for swapping records

numRecords: .word 10 # For added flexibility

recordText: .asciiz "Record"

namePrompt: .asciiz "Enter name: "

agePrompt: .asciiz "Enter age: "

salaryPrompt: .asciiz "Enter salary: "

firstRecPrompt: .asciiz "First record to swap: "

secRecPrompt: .asciiz "Second record to swap: "

invalidText: .asciiz "Invalid record"

topLineText: .asciiz "\n| R# | Name | Age | Salary "

space: .asciiz " "

newLine: .asciiz "\n"

vertLine: .asciiz "|"

horLine: .asciiz "-"

.globl main

.text

main:

jal getRecords

jal printRecords

jal swapRecords

jal printRecords

b exit

##################### INPUT CODE BLOCK BEGIN #########################

getRecords:

la $t0, recordData

lw $t1, numRecords # Number of records

li $t2, 1 # Record counter for aesthetic purposes

getRecLoop:

la $a0, recordText

li $v0, 4

syscall

la $a0, space

syscall

move $a0, $t2

li $v0, 1

syscall

la $a0, newLine

li $v0, 4

syscall

la $a0, namePrompt

syscall

move $a0, $t0

li $a1, 40

li $v0, 8

syscall

la $a0, agePrompt

li $v0, 4

syscall

li $v0, 5

syscall

sw $v0, 40( $t0 )

la $a0, salaryPrompt

li $v0, 4

syscall

li $v0, 5

syscall

sw $v0, 44( $t0 )

la $a0, newLine

li $v0, 4

syscall

addi $t0, $t0, 48

addi $t1, $t1, -1

addi $t2, $t2, 1

bgtz $t1, getRecLoop

jr $ra

######################## INPUT CODE BLOCK END ########################

##################### PRINTING CODE BLOCK BEGIN ######################

printRecords:

# Purpose: Print out a chart with all records

move $s0, $ra

la $a0, newLine

li $v0, 4

syscall

jal printHorLine

la $a0, topLineText

syscall

jal printHorLine

la $t0, recordData

lw $t1, numRecords

li $t2, 1

printRecLoop:

move $a0, $t0

move $a1, $t2

jal printLine

jal printHorLine

addi $t0, $t0, 48

addi $t1, $t1, -1

addi $t2, $t2, 1

bgtz $t1, printRecLoop

la $a0, newLine

li $v0, 4

syscall

jr $s0

printLine:

move $s1, $ra

move $t3, $a0 # Current record address

la $a0, newLine

li $v0, 4

syscall

la $a0, vertLine

syscall

la $a0, space

syscall

# Printing record number

move $a0, $a1

li $v0, 1

syscall

bgt $a1, 9, singleCont

la $a0, space

li $v0, 4

syscall # Print extra space if single digit

singleCont: jal printVertLine # Prints " | "

# Printing name

jal printFillName

jal printVertLine

# Printing Age

lw $a0, 40( $t3 )

jal printNum

jal printVertLine

# Printing Salary

lw $a0, 44( $t3 )

jal printNum

la $a0, newLine

li $v0, 4

syscall

jr $s1

printNum:

# Maintains chart structure for age/salary

li $v0, 1

move $t4, $a0

syscall

bgt $t4, 99, cont # greater than 100, no extra spaces

la $a0, space

li $v0, 4

syscall

bgt $t4, 9, cont # 10 - 99 only print one space

syscall # 0 - 9 Print extra space to align

cont: jr $ra

printFillName:

# Purpose: So for the chart I wanted the text to look formatted, so this method changes the spaces needed to maintain chart structure

move $s2, $ra

move $t4, $t3

jal removeNewline

move $t4, $t3

li $t5, 0 # Length of string

li $t7, 40 # needed to subtract

nameLengthLoop: # Get length of string

lb $t6, 0( $t4 )

beqz $t6, fillNameCont

addi $t4, $t4, 1

addi $t5, $t5, 1

b nameLengthLoop

fillNameCont: sub $t5, $t7, $t5 # Number of spaces needed to fill, 40 - length = extra space left

la $a0, 0( $t3 )

li $v0, 4

syscall

la $a0, space

nameSpaceLoop:

beqz $t5, nameSpaceCont

syscall

addi $t5, $t5, -1

b nameSpaceLoop

nameSpaceCont: jr $s2

removeNewline:

# Newline in name string was causing formatting issues so we get rid of it

li $t6, 0

removeLoop:

lb $t5, 0( $t4 )

beq $t5, 10, removeCont

addi $t4, $t4, 1

addi $t6, $t6, 1

beq $t6, 40, removeReturn # If no newline char, jump back

b removeLoop

removeCont: add $t3, $t3, $t6

sb $0, 0( $t3 ) # Replace \n with null

sub $t3, $t3, $t6

removeReturn: jr $ra

printVertLine:

# Purpose: Prints " | ", made to save typing

la $a0, space

li $v0, 4

syscall

la $a0, vertLine

syscall

la $a0, space

syscall

jr $ra

printHorLine:

# Purpose: Aesthetic, prints out the dividing lines in chart

li $t3, 68

la $a0, horLine

li $v0, 4

horLineLoop:

syscall

addi $t3, $t3, -1

bgtz $t3, horLineLoop

jr $ra

####################### PRINTING CODE BLOCK END ######################

#################### SWAP RECORD CODE BLOCK BEGIN ####################

swapRecords:

move $s0, $ra

jal getSwaps

move $t0, $v1 # First record index

move $t1, $v0 # Second record index

la $t2, recordData # First record pointer

la $t3, recordData # Second record pointer

la $t4, tmpRecord # Temp Record holder

# Set up first record pointer

addi $t0, $t0, -1 # Convert our index to 0-base

mul $t0, $t0, 48 # Determine offset for record we need

add $t2, $t2, $t0 # Move pointer to correct record

# Set up second record pointer

addi $t1, $t1, -1

mul $t1, $t1, 48

add $t3, $t3, $t1

move $a0, $t2

move $a1, $t4

jal copyRecord

move $a0, $t3

move $a1, $t2

jal copyRecord

move $a0, $t4

move $a1, $t3

jal copyRecord

jr $s0

copyRecord:

# Purpose: Copy $a0 -> $a1

li $t5, 12 # Loop counter

copyLoop:

lw $t6, 0( $a0 )

sw $t6, 0( $a1 )

addi $a0, $a0, 4

addi $a1, $a1, 4

addi $t5, $t5, -1

bgtz $t5, copyLoop

jr $ra

getSwaps:

# Purpose: Get 2 record indexes to swap

lw $t0, numRecords

inv1: la $a0, newLine

li $v0, 4

syscall

la $a0, firstRecPrompt

syscall

li $v0, 5

syscall

bltz $v0, invalid1

bgt $v0, $t0, invalid1

move $v1, $v0

inv2: la $a0, secRecPrompt

li $v0, 4

syscall

li $v0, 5

syscall

bltz $v0, invalid2

bgt $v0, $t0, invalid2

jr $ra

# Some error handling in case, user inputs unavailable records

invalid1:

la $a0, invalidText

li $v0, 4

syscall

b inv1

invalid2:

la $a0, invalidText

li $v0, 4

syscall

la $a0, newLine

syscall

b inv2

#################### SWAP RECORD CODE BLOCK END ######################

exit:

li $v0, 10

syscall