.data

array: .space 40

minWords: .asciiz "\nThe minimum of the array is "

combWord: .asciiz "\nThe combination is "

promptInt:.asciiz "Enter an integer: "

promptLH:.asciiz "Enter an index(0-9): "

nL: .asciiz "\n"

.text

main:li $t0,10

la $t1,array

loadArray:

blez $t0,doMin

li $v0,4

la $a0,promptInt

syscall

li $v0,5

syscall

sw $v0,0($t1)

addi $t0,$t0,-1

addi $t1,$t1,4

b loadArray

doMin:

li $v0,4

la $a0,promptLH

syscall

li $v0,5

syscall

move $a1,$v0

li $v0,4

la $a0,promptLH

syscall

li $v0,5

syscall

move $a2,$v0

li $v0,4

la $a0,minWords

syscall

la $a0,array

jal Min

move $t3,$v0

li $v0,1

move $a0,$t3

syscall

li $v0,4

la $a0,nL

syscall

#user input

li $v0,4

la $a0,promptInt

syscall

li $v0,5

syscall

move $t1,$v0

li $v0,4

la $a0,promptInt

syscall

li $v0,5

syscall

move $t2,$v0

li $v0,4

la $a0,combWord

syscall

move $a0,$t1

move $a1,$t2

jal Comb

move $t3,$v0

li $v0,1

move $a0,$t3

syscall

li $v0,4

la $a0,nL

syscall

li $v0,10 #exit

syscall

Comb:#a0 = n, a1 = r

beq $a0,$a1,base #r == n

beqz $a1,base#r == 0

recC:

addiu $sp,$sp,-16 #save local variables

sw $ra,0($sp) #save stack pointer

sw $a0,4($sp) #save n

sw $a1,8($sp) #save r

#Comb(n-1,r)

addi $a0,$a0,-1 #n-1

jal Comb

sw $v0,12($sp)

#Comb(n-1,r-1)

lw $a0,4($sp) #load n

addi $a0,$a0,-1 #n-1

lw $a1,8($sp) #load r

addi $a1,$a1,-1 #r-1 is in a1

jal Comb

lw $t1,12($sp) #t1 = Comb(n-1,r)

move $t2,$v0

add $v0,$t2,$t1 #Comb(n,r) = Comb(n-1,r) + Comb(n-1,r-1)

lw $ra,0($sp)

addiu $sp,$sp,16 #return space

jr $ra

base:li $v0,1

jr $ra

Min:#array address is in a0

#low is in a1, high is in a2

bne $a1,$a2,rec

mul $t0,$a1,4

add $t0,$t0,$a0

lw $v0,0($t0)

jr $ra #return

rec:add $t0,$a1,$a2#high + low #$t0 is mid

sra $t0,$t0,1 #divide by 2

addiu $sp,$sp,-16 #save local variables

sw $ra,0($sp) #save stack pointer

sw $t0,4($sp) #save mid

sw $a2,8($sp) #save high

#min1 = Min(int[]A,low,mid)

move $a2,$t0

jal Min

sw $v0,12($sp) #save min1

#min2 = Min(int[]A,mid+1,high)

lw $a1,4($sp) #mid is in a1

addi $a1,$a1,1 #mid + 1 is in a1

lw $a2,8($sp)

jal Min

move $t2,$v0 #t2 = min2

lw $t1,12($sp)#get min1 from stack

#if (min1>min2),return min2

ble $t1,$t2,retMin1

b ret #$v1 is already min2

retMin1:

move $v0,$t1 #return min1

ret:lw $ra,0($sp)

addiu $sp,$sp,16 #return space

jr $ra



