生成MIPS代码

.data

const\_int\_1: .space 4

const\_int\_2: .space 4

const\_int\_3: .space 4

const\_int\_4: .space 4

const\_char\_1: .space 4

const\_char\_2: .space 4

const\_char\_3: .space 4

const\_char\_4: .space 4

global\_int\_array\_1: .space 20

global\_int\_1: .space 4

global\_char\_array\_1: .space 20

global\_char\_1: .space 4

global\_int\_2: .space 4

global\_int\_array\_2: .space 20

global\_char\_2: .space 4

global\_char\_array\_2: .space 20

global\_char\_array\_3: .space 40

str0: .asciiz "take"

str1: .asciiz "from"

str2: .asciiz "to"

str3: .asciiz "!@\n#$^&\*()Qqaa123[];',./"

str4: .asciiz "!@\n#$^&\*()Qqaa123[];',./"

str5: .asciiz "!@\n#$^&\*()Qqaa123[];',./"

str6: .asciiz "INPUT of func\_ret\_int\_1:"

str7: .asciiz "OPERATE of func\_ret\_int\_1:"

str8: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

str9: .asciiz "Start testing global:"

str10: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

str11: .asciiz "Start testing return:"

str12: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

str13: .asciiz "Start testing I/O:"

str14: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

str15: .asciiz "Start testing assign & exp:"

str16: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

str17: .asciiz "Start testing if & while:"

str18: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

str19: .asciiz "Start testing switch:"

str20: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

str21: .asciiz "Start testing parameter:"

str22: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

str23: .asciiz "Start testing recursion:"

str24: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

.text

#const define: const\_int\_1

#load 12345679 to s1

add $s1, $0, $0

ori $s1, $s1, 12345679

add $t0, $0, $s1

sw $t0, const\_int\_1

#end const define: const\_int\_1

#const define: const\_int\_2

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

add $t0, $0, $s1

sw $t0, const\_int\_2

#end const define: const\_int\_2

#const define: const\_int\_3

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

add $t0, $0, $s1

sw $t0, const\_int\_3

#end const define: const\_int\_3

#const define: const\_int\_4

#load -12345679 to s1

add $s1, $0, $0

ori $s1, $s1, -12345679

add $t0, $0, $s1

sw $t0, const\_int\_4

#end const define: const\_int\_4

#const define: const\_char\_1

#load '9' to s1

add $s1, $0, $0

ori $s1, $s1, 57

add $t0, $0, $s1

sw $t0, const\_char\_1

#end const define: const\_char\_1

#const define: const\_char\_2

#load '\_' to s1

add $s1, $0, $0

ori $s1, $s1, 95

add $t0, $0, $s1

sw $t0, const\_char\_2

#end const define: const\_char\_2

#const define: const\_char\_3

#load '+' to s1

add $s1, $0, $0

ori $s1, $s1, 43

add $t0, $0, $s1

sw $t0, const\_char\_3

#end const define: const\_char\_3

#const define: const\_char\_4

#load '\*' to s1

add $s1, $0, $0

ori $s1, $s1, 42

add $t0, $0, $s1

sw $t0, const\_char\_4

#end const define: const\_char\_4

j main

hanoi:

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

ble $s1, $s2, \_label\_0

#sub: @temp0 = n - 1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

sub $s3, $s1, $s2

#write s3 to @temp0

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp0 = n - 1

#push: @temp0

#load @temp0 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp0 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp0

#push: from

#load from to s1

add $t0, $0, $0

ori $t0, $t0, 12

add $t2, $s0, $t0

lw $s1, ($t2)

#load from to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: from

#push: to

#load to to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load to to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: to

#push: tmp

#load tmp to s1

add $t0, $0, $0

ori $t0, $t0, 8

add $t2, $s0, $t0

lw $s1, ($t2)

#load tmp to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: tmp

#call: hanoi

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal hanoi

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: hanoi

#write string: "take"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str0

syscall

#end write string: "take"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: n

add $v0, $0, $0

ori $v0, $v0, 1

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

add $a0, $0, $s1

syscall

#end write int: n

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "from"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str1

syscall

#end write string: "from"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load from to s1

add $t0, $0, $0

ori $t0, $t0, 12

add $t2, $s0, $t0

lw $s1, ($t2)

#load from to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "to"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str2

syscall

#end write string: "to"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load to to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#sub: @temp1 = n - 1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

sub $s3, $s1, $s2

#write s3 to @temp1

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp1 = n - 1

#push: @temp1

#load @temp1 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp1 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp1

#push: tmp

#load tmp to s1

add $t0, $0, $0

ori $t0, $t0, 8

add $t2, $s0, $t0

lw $s1, ($t2)

#load tmp to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: tmp

#push: from

#load from to s1

add $t0, $0, $0

ori $t0, $t0, 12

add $t2, $s0, $t0

lw $s1, ($t2)

#load from to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: from

#push: to

#load to to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load to to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: to

#call: hanoi

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal hanoi

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: hanoi

\_label\_0:

#return:

addi $sp, $s0, 16

jr $ra

#end return:

#return:

addi $sp, $s0, 16

jr $ra

#end return:

j main

Fibonacci:

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

bne $s1, $s2, \_label\_1

#return: 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: 0

\_label\_1:

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

bne $s1, $s2, \_label\_2

#return: 1

#load 1 to s1

add $s1, $0, $0

ori $s1, $s1, 1

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: 1

\_label\_2:

#sub: @temp2 = n - 1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

sub $s3, $s1, $s2

#write s3 to @temp2

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp2 = n - 1

#push: @temp2

#load @temp2 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp2 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp2

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#sub: @temp3 = n - 2

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

sub $s3, $s1, $s2

#write s3 to @temp3

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp3 = n - 2

#push: @temp3

#load @temp3 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp3 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp3

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#add: @temp4 = @RET2 + @RET3

#load @RET2 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET2 to s1

#load @RET3 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET3 to s2

add $s3, $s1, $s2

#write s3 to @temp4

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp4 = @RET2 + @RET3

#return: @temp4

#load @temp4 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp4 to s1

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: @temp4

#return:

addi $sp, $s0, 4

jr $ra

#end return:

j main

fac:

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

bne $s1, $s2, \_label\_3

#return: 1

#load 1 to s1

add $s1, $0, $0

ori $s1, $s1, 1

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: 1

\_label\_3:

#sub: @temp5 = n - 1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

sub $s3, $s1, $s2

#write s3 to @temp5

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp5 = n - 1

#push: @temp5

#load @temp5 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp5 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp5

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#mul: @temp6 = n \* @RET4

#load n to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load n to s1

#load @RET4 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp6

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp6 = n \* @RET4

#return: @temp6

#load @temp6 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp6 to s1

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: @temp6

#return:

addi $sp, $s0, 4

jr $ra

#end return:

j main

initGlobalArray:

#assign: global\_int\_array\_1[0] = 0

addi $t0, $0, 0x10010000

addi $s2, $t0, 32

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

add $t1, $s2, $0

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_1[0] = 0

#assign: global\_int\_array\_1[1] = 1

addi $t0, $0, 0x10010000

addi $s2, $t0, 32

#load 1 to s1

add $s1, $0, $0

ori $s1, $s1, 1

add $t1, $s2, $0

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_1[1] = 1

#get: = global\_int\_array\_1[0]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[0]

#get: = global\_int\_array\_1[1]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[1]

#add: @temp7 = +

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load to s2

add $s3, $s1, $s2

#write s3 to @temp7

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp7 = +

#assign: global\_int\_array\_1[2] = @temp7

addi $t0, $0, 0x10010000

addi $s2, $t0, 32

#load @temp7 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp7 to s1

add $t1, $s2, $0

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_1[2] = @temp7

#get: = global\_int\_array\_1[2]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[2]

#get: = global\_int\_array\_1[1]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[1]

#add: @temp8 = +

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load to s2

add $s3, $s1, $s2

#write s3 to @temp8

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp8 = +

#assign: global\_int\_array\_1[3] = @temp8

addi $t0, $0, 0x10010000

addi $s2, $t0, 32

#load @temp8 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp8 to s1

add $t1, $s2, $0

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_1[3] = @temp8

#get: = global\_int\_array\_1[3]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[3]

#get: = global\_int\_array\_1[2]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[2]

#add: @temp9 = +

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load to s2

add $s3, $s1, $s2

#write s3 to @temp9

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp9 = +

#assign: global\_int\_array\_1[4] = @temp9

addi $t0, $0, 0x10010000

addi $s2, $t0, 32

#load @temp9 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp9 to s1

add $t1, $s2, $0

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_1[4] = @temp9

#get: = global\_int\_array\_1[1]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[1]

#assign: global\_int\_array\_2[0] =

addi $t0, $0, 0x10010000

addi $s2, $t0, 84

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $t1, $s2, $0

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_2[0] =

#get: = global\_int\_array\_2[0]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[0]

#get: = global\_int\_array\_1[2]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[2]

#mul: @temp10 = \*

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp10

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp10 = \*

#assign: global\_int\_array\_2[1] = @temp10

addi $t0, $0, 0x10010000

addi $s2, $t0, 84

#load @temp10 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp10 to s1

add $t1, $s2, $0

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_2[1] = @temp10

#get: = global\_int\_array\_2[1]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[1]

#get: = global\_int\_array\_1[3]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[3]

#mul: @temp11 = \*

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp11

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp11 = \*

#assign: global\_int\_array\_2[2] = @temp11

addi $t0, $0, 0x10010000

addi $s2, $t0, 84

#load @temp11 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp11 to s1

add $t1, $s2, $0

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_2[2] = @temp11

#get: = global\_int\_array\_2[2]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[2]

#get: = global\_int\_array\_1[4]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[4]

#mul: @temp12 = \*

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp12

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp12 = \*

#assign: global\_int\_array\_2[3] = @temp12

addi $t0, $0, 0x10010000

addi $s2, $t0, 84

#load @temp12 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp12 to s1

add $t1, $s2, $0

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_2[3] = @temp12

#get: = global\_int\_array\_2[3]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[3]

#get: = global\_int\_array\_1[4]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[4]

#div: @temp13 = /

#load to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load to s2

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

div $s1, $s2

mflo $s3

#write s3 to @temp13

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end div: @temp13 = /

#assign: global\_int\_array\_2[4] = @temp13

addi $t0, $0, 0x10010000

addi $s2, $t0, 84

#load @temp13 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp13 to s1

add $t1, $s2, $0

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_int\_array\_2[4] = @temp13

#assign: global\_char\_array\_1[0] = 'a'

addi $t0, $0, 0x10010000

addi $s2, $t0, 56

#load 'a' to s1

add $s1, $0, $0

ori $s1, $s1, 97

add $t1, $s2, $0

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_1[0] = 'a'

#assign: global\_char\_array\_1[1] = 'A'

addi $t0, $0, 0x10010000

addi $s2, $t0, 56

#load 'A' to s1

add $s1, $0, $0

ori $s1, $s1, 65

add $t1, $s2, $0

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_1[1] = 'A'

#assign: global\_char\_array\_1[2] = 'z'

addi $t0, $0, 0x10010000

addi $s2, $t0, 56

#load 'z' to s1

add $s1, $0, $0

ori $s1, $s1, 122

add $t1, $s2, $0

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_1[2] = 'z'

#assign: global\_char\_array\_1[3] = 'Z'

addi $t0, $0, 0x10010000

addi $s2, $t0, 56

#load 'Z' to s1

add $s1, $0, $0

ori $s1, $s1, 90

add $t1, $s2, $0

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_1[3] = 'Z'

#assign: global\_char\_array\_1[4] = '\_'

addi $t0, $0, 0x10010000

addi $s2, $t0, 56

#load '\_' to s1

add $s1, $0, $0

ori $s1, $s1, 95

add $t1, $s2, $0

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_1[4] = '\_'

#assign: global\_char\_array\_2[0] = '+'

addi $t0, $0, 0x10010000

addi $s2, $t0, 108

#load '+' to s1

add $s1, $0, $0

ori $s1, $s1, 43

add $t1, $s2, $0

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_2[0] = '+'

#assign: global\_char\_array\_2[1] = '-'

addi $t0, $0, 0x10010000

addi $s2, $t0, 108

#load '-' to s1

add $s1, $0, $0

ori $s1, $s1, 45

add $t1, $s2, $0

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_2[1] = '-'

#assign: global\_char\_array\_2[2] = '\*'

addi $t0, $0, 0x10010000

addi $s2, $t0, 108

#load '\*' to s1

add $s1, $0, $0

ori $s1, $s1, 42

add $t1, $s2, $0

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_2[2] = '\*'

#assign: global\_char\_array\_2[3] = '/'

addi $t0, $0, 0x10010000

addi $s2, $t0, 108

#load '/' to s1

add $s1, $0, $0

ori $s1, $s1, 47

add $t1, $s2, $0

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_2[3] = '/'

#assign: global\_char\_array\_2[4] = '6'

addi $t0, $0, 0x10010000

addi $s2, $t0, 108

#load '6' to s1

add $s1, $0, $0

ori $s1, $s1, 54

add $t1, $s2, $0

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_2[4] = '6'

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

assignGlobal:

#assign: global\_int\_1 = a

#load a to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load a to s1

add $s3, $0, $s1

#write s3 to global\_int\_1

addi $t0, $0, 0x10010000

addi $t0, $t0, 52

sw $s3, ($t0)

#end write s3

#end assign: global\_int\_1 = a

#assign: global\_char\_1 = b

#load b to s1

add $t0, $0, $0

ori $t0, $t0, 12

add $t2, $s0, $t0

lw $s1, ($t2)

#load b to s1

add $s3, $0, $s1

#write s3 to global\_char\_1

addi $t0, $0, 0x10010000

addi $t0, $t0, 76

sw $s3, ($t0)

#end write s3

#end assign: global\_char\_1 = b

#assign: global\_int\_2 = c

#load c to s1

add $t0, $0, $0

ori $t0, $t0, 8

add $t2, $s0, $t0

lw $s1, ($t2)

#load c to s1

add $s3, $0, $s1

#write s3 to global\_int\_2

addi $t0, $0, 0x10010000

addi $t0, $t0, 80

sw $s3, ($t0)

#end write s3

#end assign: global\_int\_2 = c

#assign: global\_char\_2 = d

#load d to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load d to s1

add $s3, $0, $s1

#write s3 to global\_char\_2

addi $t0, $0, 0x10010000

addi $t0, $t0, 104

sw $s3, ($t0)

#end write s3

#end assign: global\_char\_2 = d

#return:

addi $sp, $s0, 16

jr $ra

#end return:

#return:

addi $sp, $s0, 16

jr $ra

#end return:

j main

printGlobalConst:

#write int: const\_int\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load const\_int\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 0

lw $s1, ($t0)

#load const\_int\_1 to s1

add $a0, $0, $s1

syscall

#end write int: const\_int\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: const\_int\_2

add $v0, $0, $0

ori $v0, $v0, 1

#load const\_int\_2 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 4

lw $s1, ($t0)

#load const\_int\_2 to s1

add $a0, $0, $s1

syscall

#end write int: const\_int\_2

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: const\_int\_3

add $v0, $0, $0

ori $v0, $v0, 1

#load const\_int\_3 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 8

lw $s1, ($t0)

#load const\_int\_3 to s1

add $a0, $0, $s1

syscall

#end write int: const\_int\_3

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: const\_int\_4

add $v0, $0, $0

ori $v0, $v0, 1

#load const\_int\_4 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s1, ($t0)

#load const\_int\_4 to s1

add $a0, $0, $s1

syscall

#end write int: const\_int\_4

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load const\_char\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 16

lw $s1, ($t0)

#load const\_char\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load const\_char\_2 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 20

lw $s1, ($t0)

#load const\_char\_2 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load const\_char\_3 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 24

lw $s1, ($t0)

#load const\_char\_3 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load const\_char\_4 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 28

lw $s1, ($t0)

#load const\_char\_4 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

printGlobalVar:

#write int: global\_int\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load global\_int\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 52

lw $s1, ($t0)

#load global\_int\_1 to s1

add $a0, $0, $s1

syscall

#end write int: global\_int\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: global\_int\_2

add $v0, $0, $0

ori $v0, $v0, 1

#load global\_int\_2 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 80

lw $s1, ($t0)

#load global\_int\_2 to s1

add $a0, $0, $s1

syscall

#end write int: global\_int\_2

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load global\_char\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 76

lw $s1, ($t0)

#load global\_char\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load global\_char\_2 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 104

lw $s1, ($t0)

#load global\_char\_2 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

printGlobalArray:

#get: = global\_int\_array\_1[0]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[0]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_1[1]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[1]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_1[2]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[2]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_1[3]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[3]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_1[4]

addi $t0, $0, 0x10010000

addi $s1, $t0, 32

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_1[4]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_2[0]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[0]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_2[1]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[1]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_2[2]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[2]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_2[3]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[3]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_int\_array\_2[4]

addi $t0, $0, 0x10010000

addi $s1, $t0, 84

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_int\_array\_2[4]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_1[0]

addi $t0, $0, 0x10010000

addi $s1, $t0, 56

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_1[0]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_1[1]

addi $t0, $0, 0x10010000

addi $s1, $t0, 56

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_1[1]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_1[2]

addi $t0, $0, 0x10010000

addi $s1, $t0, 56

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_1[2]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_1[3]

addi $t0, $0, 0x10010000

addi $s1, $t0, 56

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_1[3]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_1[4]

addi $t0, $0, 0x10010000

addi $s1, $t0, 56

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_1[4]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_2[0]

addi $t0, $0, 0x10010000

addi $s1, $t0, 108

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_2[0]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_2[1]

addi $t0, $0, 0x10010000

addi $s1, $t0, 108

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_2[1]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_2[2]

addi $t0, $0, 0x10010000

addi $s1, $t0, 108

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_2[2]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_2[3]

addi $t0, $0, 0x10010000

addi $s1, $t0, 108

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_2[3]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = global\_char\_array\_2[4]

addi $t0, $0, 0x10010000

addi $s1, $t0, 108

#load 4 to s2

add $s2, $0, $0

ori $s2, $s2, 4

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_2[4]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

testGlobal:

#call: initGlobalArray

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal initGlobalArray

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: initGlobalArray

#push: const\_int\_1

#load const\_int\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 0

lw $s1, ($t0)

#load const\_int\_1 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: const\_int\_1

#push: const\_char\_1

#load const\_char\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 16

lw $s1, ($t0)

#load const\_char\_1 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: const\_char\_1

#push: const\_int\_4

#load const\_int\_4 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s1, ($t0)

#load const\_int\_4 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: const\_int\_4

#push: const\_char\_4

#load const\_char\_4 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 28

lw $s1, ($t0)

#load const\_char\_4 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: const\_char\_4

#call: assignGlobal

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal assignGlobal

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: assignGlobal

#call: printGlobalConst

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal printGlobalConst

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: printGlobalConst

#call: printGlobalVar

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal printGlobalVar

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: printGlobalVar

#call: printGlobalArray

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal printGlobalArray

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: printGlobalArray

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

testReturnInt:

#add: @temp14 = a + 1

#load a to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load a to s1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

add $s3, $s1, $s2

#write s3 to @temp14

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp14 = a + 1

#return: @temp14

#load @temp14 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp14 to s1

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: @temp14

#return:

addi $sp, $s0, 4

jr $ra

#end return:

j main

testReturnChar:

#load a to s4

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s4, ($t2)

#load a to s4

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

bne $s4, $s2, \_label\_5

#return: 'a'

#load 'a' to s1

add $s1, $0, $0

ori $s1, $s1, 97

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: 'a'

j \_label\_4

\_label\_5:

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

bne $s4, $s2, \_label\_6

#return: 'b'

#load 'b' to s1

add $s1, $0, $0

ori $s1, $s1, 98

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: 'b'

j \_label\_4

\_label\_6:

#load 3 to s2

add $s2, $0, $0

ori $s2, $s2, 3

bne $s4, $s2, \_label\_7

#return: 'c'

#load 'c' to s1

add $s1, $0, $0

ori $s1, $s1, 99

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: 'c'

j \_label\_4

\_label\_7:

j \_label\_4

\_label\_4:

#return: '\_'

#load '\_' to s1

add $s1, $0, $0

ori $s1, $s1, 95

add $v0, $0, $s1

addi $sp, $s0, 4

jr $ra

#end return: '\_'

#return:

addi $sp, $s0, 4

jr $ra

#end return:

j main

testReturn:

#push: 3

#load 3 to s1

add $s1, $0, $0

ori $s1, $s1, 3

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 3

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#push: 2

#load 2 to s1

add $s1, $0, $0

ori $s1, $s1, 2

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 2

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#add: @temp15 = @RET10 + @RET11

#load @RET10 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET10 to s1

#load @RET11 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET11 to s2

add $s3, $s1, $s2

#write s3 to @temp15

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp15 = @RET10 + @RET11

#push: @temp15

#load @temp15 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp15 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp15

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#push: @RET12

#load @RET12 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET12 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @RET12

#call: testReturnInt

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnInt

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnInt

#write int: @RET13

add $v0, $0, $0

ori $v0, $v0, 1

#load @RET13 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET13 to s1

add $a0, $0, $s1

syscall

#end write int: @RET13

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#push: 1

#load 1 to s1

add $s1, $0, $0

ori $s1, $s1, 1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 1

#call: testReturnChar

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnChar

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnChar

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load @RET14 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET14 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#push: 2

#load 2 to s1

add $s1, $0, $0

ori $s1, $s1, 2

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 2

#call: testReturnChar

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnChar

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnChar

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load @RET15 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET15 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#push: 3

#load 3 to s1

add $s1, $0, $0

ori $s1, $s1, 3

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 3

#call: testReturnChar

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnChar

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnChar

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load @RET16 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET16 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#push: 4

#load 4 to s1

add $s1, $0, $0

ori $s1, $s1, 4

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 4

#call: testReturnChar

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnChar

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnChar

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load @RET17 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET17 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

testAssignAndExp:

#const define: const\_int\_1

#load 1 to s1

add $s1, $0, $0

ori $s1, $s1, 1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end const define: const\_int\_1

#const define: const\_int\_2

#load -1 to s1

add $s1, $0, $0

ori $s1, $s1, -1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end const define: const\_int\_2

#const define: const\_int\_3

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

sw $s1, ($sp)

addiu $sp, $sp, -4

#end const define: const\_int\_3

#const define: const\_char\_1

#load 'a' to s1

add $s1, $0, $0

ori $s1, $s1, 97

sw $s1, ($sp)

addiu $sp, $sp, -4

#end const define: const\_char\_1

#const define: const\_char\_2

#load 'A' to s1

add $s1, $0, $0

ori $s1, $s1, 65

sw $s1, ($sp)

addiu $sp, $sp, -4

#end const define: const\_char\_2

#const define: const\_char\_3

#load 'b' to s1

add $s1, $0, $0

ori $s1, $s1, 98

sw $s1, ($sp)

addiu $sp, $sp, -4

#end const define: const\_char\_3

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

#push: 5

#load 5 to s1

add $s1, $0, $0

ori $s1, $s1, 5

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 5

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#push: @RET18

#load @RET18 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET18 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @RET18

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#push: @RET19

#load @RET19 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET19 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @RET19

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#mul: @temp16 = const\_int\_2 \* const\_int\_4

#load const\_int\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_2 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp16

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp16 = const\_int\_2 \* const\_int\_4

#div: @temp17 = const\_int\_1 / @temp16

#load @temp16 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp16 to s2

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

div $s1, $s2

mflo $s3

#write s3 to @temp17

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end div: @temp17 = const\_int\_1 / @temp16

#add: @temp18 = @RET20 + @temp17

#load @RET20 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET20 to s1

#load @temp17 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp17 to s2

add $s3, $s1, $s2

#write s3 to @temp18

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp18 = @RET20 + @temp17

#push: 3

#load 3 to s1

add $s1, $0, $0

ori $s1, $s1, 3

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 3

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#div: @temp19 = @temp18 / @RET21

#load @RET21 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET21 to s2

#load @temp18 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp18 to s1

div $s1, $s2

mflo $s3

#write s3 to @temp19

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end div: @temp19 = @temp18 / @RET21

#add: @temp20 = const\_int\_1 + @temp19

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

#load @temp19 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp19 to s2

add $s3, $s1, $s2

#write s3 to @temp20

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp20 = const\_int\_1 + @temp19

#assign: int\_temp\_1 = @temp20

#load @temp20 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp20 to s1

add $s3, $0, $s1

#write s3 to int\_temp\_1

add $t0, $0, $0

ori $t0, $t0, 32

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: int\_temp\_1 = @temp20

#push: 3

#load 3 to s1

add $s1, $0, $0

ori $s1, $s1, 3

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 3

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#push: 2

#load 2 to s1

add $s1, $0, $0

ori $s1, $s1, 2

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 2

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#add: @temp21 = @RET22 + @RET23

#load @RET22 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET22 to s1

#load @RET23 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET23 to s2

add $s3, $s1, $s2

#write s3 to @temp21

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp21 = @RET22 + @RET23

#push: @temp21

#load @temp21 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp21 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp21

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#assign: int\_temp\_2 = @RET24

#load @RET24 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET24 to s1

add $s3, $0, $s1

#write s3 to int\_temp\_2

add $t0, $0, $0

ori $t0, $t0, 36

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: int\_temp\_2 = @RET24

#push: 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 0

#call: testReturnInt

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnInt

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnInt

#assign: int\_array[const\_int\_3] = @RET25

subi $s2, $s0, 40

#load @RET25 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET25 to s1

add $t1, $s2, $0

#load const\_int\_3 to s2

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s2, ($t0)

#load const\_int\_3 to s2

sll $s2, $s2, 2

sub $t0, $t1, $s2

sw $s1, ($t0)

#end assign: int\_array[const\_int\_3] = @RET25

#mul: @temp22 = const\_int\_2 \* const\_int\_4

#load const\_int\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_2 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp22

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp22 = const\_int\_2 \* const\_int\_4

#push: 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 0

#call: testReturnInt

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnInt

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnInt

#mul: @temp23 = @RET26 \* const\_int\_4

#load @RET26 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET26 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp23

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp23 = @RET26 \* const\_int\_4

#add: @temp24 = @temp22 + @temp23

#load @temp22 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp22 to s1

#load @temp23 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp23 to s2

add $s3, $s1, $s2

#write s3 to @temp24

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp24 = @temp22 + @temp23

#sub: @temp25 = const\_int\_1 - @temp24

#load @temp24 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp24 to s2

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

sub $s3, $s1, $s2

#write s3 to @temp25

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp25 = const\_int\_1 - @temp24

#push: const\_int\_1

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: const\_int\_1

#call: testReturnInt

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnInt

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnInt

#add: @temp26 = const\_int\_1 + @RET27

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

#load @RET27 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET27 to s2

add $s3, $s1, $s2

#write s3 to @temp26

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp26 = const\_int\_1 + @RET27

#assign: int\_array[@temp25] = @temp26

subi $s2, $s0, 40

#load @temp26 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp26 to s1

add $t1, $s2, $0

#load @temp25 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp25 to s2

sll $s2, $s2, 2

sub $t0, $t1, $s2

sw $s1, ($t0)

#end assign: int\_array[@temp25] = @temp26

#get: = int\_array[const\_int\_3]

subi $s1, $s0, 40

#load const\_int\_3 to s2

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s2, ($t0)

#load const\_int\_3 to s2

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = int\_array[const\_int\_3]

#mul: @temp27 = const\_int\_2 \* const\_int\_4

#load const\_int\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_2 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp27

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp27 = const\_int\_2 \* const\_int\_4

#push: 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 0

#call: testReturnInt

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnInt

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnInt

#mul: @temp28 = @RET28 \* const\_int\_4

#load @RET28 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET28 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp28

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp28 = @RET28 \* const\_int\_4

#add: @temp29 = @temp27 + @temp28

#load @temp27 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp27 to s1

#load @temp28 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp28 to s2

add $s3, $s1, $s2

#write s3 to @temp29

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp29 = @temp27 + @temp28

#sub: @temp30 = const\_int\_1 - @temp29

#load @temp29 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp29 to s2

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

sub $s3, $s1, $s2

#write s3 to @temp30

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp30 = const\_int\_1 - @temp29

#get: = int\_array[@temp30]

subi $s1, $s0, 40

#load @temp30 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp30 to s2

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = int\_array[@temp30]

#sub: @temp31 = 'b' - 'a'

#load 'a' to s2

add $s2, $0, $0

ori $s2, $s2, 97

#load 'b' to s1

add $s1, $0, $0

ori $s1, $s1, 98

sub $s3, $s1, $s2

#write s3 to @temp31

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp31 = 'b' - 'a'

#mul: @temp32 = \* @temp31

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load @temp31 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp31 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp32

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp32 = \* @temp31

#sub: @temp33 = - @temp32

#load @temp32 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp32 to s2

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

sub $s3, $s1, $s2

#write s3 to @temp33

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp33 = - @temp32

#assign: int\_temp\_3 = @temp33

#load @temp33 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp33 to s1

add $s3, $0, $s1

#write s3 to int\_temp\_3

add $t0, $0, $0

ori $t0, $t0, 48

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: int\_temp\_3 = @temp33

#get: = int\_array[0]

subi $s1, $s0, 40

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = int\_array[0]

#push:

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push:

#call: testReturnChar

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnChar

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnChar

#assign: char\_array[0] = @RET29

subi $s2, $s0, 56

#load @RET29 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET29 to s1

add $t1, $s2, $0

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

sub $t0, $t1, $s2

sw $s1, ($t0)

#end assign: char\_array[0] = @RET29

#mul: @temp34 = const\_int\_2 \* const\_int\_4

#load const\_int\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_2 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp34

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp34 = const\_int\_2 \* const\_int\_4

#push: 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 0

#call: testReturnInt

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnInt

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnInt

#mul: @temp35 = @RET30 \* const\_int\_4

#load @RET30 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET30 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp35

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp35 = @RET30 \* const\_int\_4

#add: @temp36 = @temp34 + @temp35

#load @temp34 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp34 to s1

#load @temp35 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp35 to s2

add $s3, $s1, $s2

#write s3 to @temp36

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp36 = @temp34 + @temp35

#sub: @temp37 = const\_int\_1 - @temp36

#load @temp36 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp36 to s2

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

sub $s3, $s1, $s2

#write s3 to @temp37

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp37 = const\_int\_1 - @temp36

#get: = int\_array[1]

subi $s1, $s0, 40

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = int\_array[1]

#add: @temp38 = + const\_int\_1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load const\_int\_1 to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load const\_int\_1 to s2

add $s3, $s1, $s2

#write s3 to @temp38

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp38 = + const\_int\_1

#push: @temp38

#load @temp38 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp38 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp38

#call: testReturnChar

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnChar

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnChar

#assign: char\_array[@temp37] = @RET31

subi $s2, $s0, 56

#load @RET31 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET31 to s1

add $t1, $s2, $0

#load @temp37 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp37 to s2

sll $s2, $s2, 2

sub $t0, $t1, $s2

sw $s1, ($t0)

#end assign: char\_array[@temp37] = @RET31

#get: = char\_array[0]

subi $s1, $s0, 56

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = char\_array[0]

#assign: char\_temp\_1 =

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $s3, $0, $s1

#write s3 to char\_temp\_1

add $t0, $0, $0

ori $t0, $t0, 52

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: char\_temp\_1 =

#mul: @temp39 = const\_int\_2 \* const\_int\_4

#load const\_int\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_2 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp39

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp39 = const\_int\_2 \* const\_int\_4

#push: 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 0

#call: testReturnInt

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnInt

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnInt

#mul: @temp40 = @RET32 \* const\_int\_4

#load @RET32 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET32 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp40

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp40 = @RET32 \* const\_int\_4

#add: @temp41 = @temp39 + @temp40

#load @temp39 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp39 to s1

#load @temp40 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp40 to s2

add $s3, $s1, $s2

#write s3 to @temp41

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp41 = @temp39 + @temp40

#sub: @temp42 = const\_int\_1 - @temp41

#load @temp41 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp41 to s2

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

sub $s3, $s1, $s2

#write s3 to @temp42

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp42 = const\_int\_1 - @temp41

#get: = char\_array[@temp42]

subi $s1, $s0, 56

#load @temp42 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp42 to s2

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = char\_array[@temp42]

#assign: char\_temp\_2 =

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $s3, $0, $s1

#write s3 to char\_temp\_2

add $t0, $0, $0

ori $t0, $t0, 64

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: char\_temp\_2 =

#write int: int\_temp\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load int\_temp\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 32

sub $t0, $s0, $t0

lw $s1, ($t0)

#load int\_temp\_1 to s1

add $a0, $0, $s1

syscall

#end write int: int\_temp\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: int\_temp\_2

add $v0, $0, $0

ori $v0, $v0, 1

#load int\_temp\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 36

sub $t0, $s0, $t0

lw $s1, ($t0)

#load int\_temp\_2 to s1

add $a0, $0, $s1

syscall

#end write int: int\_temp\_2

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: int\_temp\_3

add $v0, $0, $0

ori $v0, $v0, 1

#load int\_temp\_3 to s1

add $t0, $0, $0

ori $t0, $t0, 48

sub $t0, $s0, $t0

lw $s1, ($t0)

#load int\_temp\_3 to s1

add $a0, $0, $s1

syscall

#end write int: int\_temp\_3

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = int\_array[const\_int\_3]

subi $s1, $s0, 40

#load const\_int\_3 to s2

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s2, ($t0)

#load const\_int\_3 to s2

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = int\_array[const\_int\_3]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#mul: @temp43 = const\_int\_2 \* const\_int\_4

#load const\_int\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_2 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp43

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp43 = const\_int\_2 \* const\_int\_4

#push: 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 0

#call: testReturnInt

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturnInt

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturnInt

#mul: @temp44 = @RET33 \* const\_int\_4

#load @RET33 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET33 to s1

#load const\_int\_4 to s2

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s2, ($t0)

#load const\_int\_4 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp44

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp44 = @RET33 \* const\_int\_4

#add: @temp45 = @temp43 + @temp44

#load @temp43 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp43 to s1

#load @temp44 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp44 to s2

add $s3, $s1, $s2

#write s3 to @temp45

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp45 = @temp43 + @temp44

#sub: @temp46 = const\_int\_1 - @temp45

#load @temp45 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp45 to s2

#load const\_int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load const\_int\_1 to s1

sub $s3, $s1, $s2

#write s3 to @temp46

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp46 = const\_int\_1 - @temp45

#get: = int\_array[@temp46]

subi $s1, $s0, 40

#load @temp46 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp46 to s2

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = int\_array[@temp46]

#write int:

add $v0, $0, $0

ori $v0, $v0, 1

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write int:

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load char\_temp\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 52

sub $t0, $s0, $t0

lw $s1, ($t0)

#load char\_temp\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load char\_temp\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 64

sub $t0, $s0, $t0

lw $s1, ($t0)

#load char\_temp\_2 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = char\_array[0]

subi $s1, $s0, 56

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = char\_array[0]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = char\_array[const\_int\_1]

subi $s1, $s0, 56

#load const\_int\_1 to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load const\_int\_1 to s2

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = char\_array[const\_int\_1]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#get: = char\_array[0]

subi $s1, $s0, 56

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = char\_array[0]

#get: = char\_array[0]

subi $s1, $s0, 56

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

sll $s2, $s2, 2

sub $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = char\_array[0]

#mul: @temp47 = \* char\_temp\_2

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load char\_temp\_2 to s2

add $t0, $0, $0

ori $t0, $t0, 64

sub $t0, $s0, $t0

lw $s2, ($t0)

#load char\_temp\_2 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp47

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp47 = \* char\_temp\_2

#add: @temp48 = + @temp47

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

#load @temp47 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp47 to s2

add $s3, $s1, $s2

#write s3 to @temp48

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp48 = + @temp47

#mul: @temp49 = char\_temp\_1 \* @temp48

#load char\_temp\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 52

sub $t0, $s0, $t0

lw $s1, ($t0)

#load char\_temp\_1 to s1

#load @temp48 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp48 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp49

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp49 = char\_temp\_1 \* @temp48

#write int: @temp49

add $v0, $0, $0

ori $v0, $v0, 1

#load @temp49 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp49 to s1

add $a0, $0, $s1

syscall

#end write int: @temp49

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

testIO:

#read int: global\_int\_1

add $v0, $0, $0

ori $v0, $v0, 5

syscall

add $s3, $0, $v0

#write s3 to global\_int\_1

addi $t0, $0, 0x10010000

addi $t0, $t0, 52

sw $s3, ($t0)

#end write s3

#end read int: global\_int\_1

#read char: global\_char\_1

add $v0, $0, $0

ori $v0, $v0, 12

syscall

add $s3, $0, $v0

#write s3 to global\_char\_1

addi $t0, $0, 0x10010000

addi $t0, $t0, 76

sw $s3, ($t0)

#end write s3

#end read char: global\_char\_1

#write string: "!@\n#$^&\*()Qqaa123[];',./"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str3

syscall

#end write string: "!@\n#$^&\*()Qqaa123[];',./"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: global\_int\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load global\_int\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 52

lw $s1, ($t0)

#load global\_int\_1 to s1

add $a0, $0, $s1

syscall

#end write int: global\_int\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load global\_char\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 76

lw $s1, ($t0)

#load global\_char\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#read int: global\_int\_1

add $v0, $0, $0

ori $v0, $v0, 5

syscall

add $s3, $0, $v0

#write s3 to global\_int\_1

addi $t0, $0, 0x10010000

addi $t0, $t0, 52

sw $s3, ($t0)

#end write s3

#end read int: global\_int\_1

#read char: global\_char\_1

add $v0, $0, $0

ori $v0, $v0, 12

syscall

add $s3, $0, $v0

#write s3 to global\_char\_1

addi $t0, $0, 0x10010000

addi $t0, $t0, 76

sw $s3, ($t0)

#end write s3

#end read char: global\_char\_1

#write string: "!@\n#$^&\*()Qqaa123[];',./"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str4

syscall

#end write string: "!@\n#$^&\*()Qqaa123[];',./"

#write int: global\_int\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load global\_int\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 52

lw $s1, ($t0)

#load global\_int\_1 to s1

add $a0, $0, $s1

syscall

#end write int: global\_int\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load global\_char\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 76

lw $s1, ($t0)

#load global\_char\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#read char: global\_char\_1

add $v0, $0, $0

ori $v0, $v0, 12

syscall

add $s3, $0, $v0

#write s3 to global\_char\_1

addi $t0, $0, 0x10010000

addi $t0, $t0, 76

sw $s3, ($t0)

#end write s3

#end read char: global\_char\_1

#read int: global\_int\_1

add $v0, $0, $0

ori $v0, $v0, 5

syscall

add $s3, $0, $v0

#write s3 to global\_int\_1

addi $t0, $0, 0x10010000

addi $t0, $t0, 52

sw $s3, ($t0)

#end write s3

#end read int: global\_int\_1

#write string: "!@\n#$^&\*()Qqaa123[];',./"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str5

syscall

#end write string: "!@\n#$^&\*()Qqaa123[];',./"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load global\_char\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 76

lw $s1, ($t0)

#load global\_char\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: global\_int\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load global\_int\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 52

lw $s1, ($t0)

#load global\_int\_1 to s1

add $a0, $0, $s1

syscall

#end write int: global\_int\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

testIfWhile:

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

#assign: k = 10

#load 10 to s1

add $s1, $0, $0

ori $s1, $s1, 10

add $s3, $0, $s1

#write s3 to k

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: k = 10

#assign: a = '+'

#load '+' to s1

add $s1, $0, $0

ori $s1, $s1, 43

add $s3, $0, $s1

#write s3 to a

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: a = '+'

#assign: b = '-'

#load '-' to s1

add $s1, $0, $0

ori $s1, $s1, 45

add $s3, $0, $s1

#write s3 to b

add $t0, $0, $0

ori $t0, $t0, 24

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: b = '-'

#assign: c = '\*'

#load '\*' to s1

add $s1, $0, $0

ori $s1, $s1, 42

add $s3, $0, $s1

#write s3 to c

add $t0, $0, $0

ori $t0, $t0, 28

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: c = '\*'

#assign: d = '/'

#load '/' to s1

add $s1, $0, $0

ori $s1, $s1, 47

add $s3, $0, $s1

#write s3 to d

add $t0, $0, $0

ori $t0, $t0, 32

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: d = '/'

#load k to s1

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s1, ($t0)

#load k to s1

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

ble $s1, $s2, \_label\_8

#load k to s1

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s1, ($t0)

#load k to s1

#load 10 to s2

add $s2, $0, $0

ori $s2, $s2, 10

bgt $s1, $s2, \_label\_9

#assign: i = 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

add $s3, $0, $s1

#write s3 to i

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: i = 0

\_label\_11:

#div: @temp50 = k / 2

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

#load k to s1

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s1, ($t0)

#load k to s1

div $s1, $s2

mflo $s3

#write s3 to @temp50

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end div: @temp50 = k / 2

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

#load @temp50 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp50 to s2

bge $s1, $s2, \_label\_10

#div: @temp51 = i / 2

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

div $s1, $s2

mflo $s3

#write s3 to @temp51

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end div: @temp51 = i / 2

#mul: @temp52 = @temp51 \* 2

#load @temp51 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp51 to s1

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

mult $s1, $s2

mflo $s3

#write s3 to @temp52

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp52 = @temp51 \* 2

#sub: @temp53 = i - @temp52

#load @temp52 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp52 to s2

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

sub $s3, $s1, $s2

#write s3 to @temp53

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp53 = i - @temp52

#load @temp53 to s4

addiu $sp, $sp, 4

lw $s4, ($sp)

#load @temp53 to s4

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

bne $s4, $s2, \_label\_13

#assign: global\_char\_array\_3[i] = a

addi $t0, $0, 0x10010000

addi $s2, $t0, 128

#load a to s1

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

lw $s1, ($t0)

#load a to s1

add $t1, $s2, $0

#load i to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load i to s2

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_3[i] = a

j \_label\_12

\_label\_13:

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

bne $s4, $s2, \_label\_14

#assign: global\_char\_array\_3[i] = b

addi $t0, $0, 0x10010000

addi $s2, $t0, 128

#load b to s1

add $t0, $0, $0

ori $t0, $t0, 24

sub $t0, $s0, $t0

lw $s1, ($t0)

#load b to s1

add $t1, $s2, $0

#load i to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load i to s2

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_3[i] = b

j \_label\_12

\_label\_14:

\_label\_12:

#get: = global\_char\_array\_3[i]

addi $t0, $0, 0x10010000

addi $s1, $t0, 128

#load i to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load i to s2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_3[i]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#add: @temp54 = i + 1

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

add $s3, $s1, $s2

#write s3 to @temp54

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp54 = i + 1

#assign: i = @temp54

#load @temp54 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp54 to s1

add $s3, $0, $s1

#write s3 to i

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: i = @temp54

j \_label\_11

\_label\_10:

\_label\_16:

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

#load k to s2

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s2, ($t0)

#load k to s2

bge $s1, $s2, \_label\_15

#div: @temp55 = i / 2

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

div $s1, $s2

mflo $s3

#write s3 to @temp55

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end div: @temp55 = i / 2

#mul: @temp56 = @temp55 \* 2

#load @temp55 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp55 to s1

#load 2 to s2

add $s2, $0, $0

ori $s2, $s2, 2

mult $s1, $s2

mflo $s3

#write s3 to @temp56

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp56 = @temp55 \* 2

#sub: @temp57 = i - @temp56

#load @temp56 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @temp56 to s2

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

sub $s3, $s1, $s2

#write s3 to @temp57

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp57 = i - @temp56

#load @temp57 to s4

addiu $sp, $sp, 4

lw $s4, ($sp)

#load @temp57 to s4

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

bne $s4, $s2, \_label\_18

#assign: global\_char\_array\_3[i] = c

addi $t0, $0, 0x10010000

addi $s2, $t0, 128

#load c to s1

add $t0, $0, $0

ori $t0, $t0, 28

sub $t0, $s0, $t0

lw $s1, ($t0)

#load c to s1

add $t1, $s2, $0

#load i to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load i to s2

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_3[i] = c

j \_label\_17

\_label\_18:

#assign: global\_char\_array\_3[i] = d

addi $t0, $0, 0x10010000

addi $s2, $t0, 128

#load d to s1

add $t0, $0, $0

ori $t0, $t0, 32

sub $t0, $s0, $t0

lw $s1, ($t0)

#load d to s1

add $t1, $s2, $0

#load i to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load i to s2

sll $s2, $s2, 2

add $t0, $t1, $s2

sw $s1, ($t0)

#end assign: global\_char\_array\_3[i] = d

j \_label\_17

\_label\_17:

#get: = global\_char\_array\_3[i]

addi $t0, $0, 0x10010000

addi $s1, $t0, 128

#load i to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load i to s2

sll $s2, $s2, 2

add $t0, $s1, $s2

lw $s3, ($t0)

#write s3 to

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end get: = global\_char\_array\_3[i]

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#add: @temp58 = i + 1

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

add $s3, $s1, $s2

#write s3 to @temp58

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp58 = i + 1

#assign: i = @temp58

#load @temp58 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp58 to s1

add $s3, $0, $s1

#write s3 to i

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: i = @temp58

j \_label\_16

\_label\_15:

\_label\_9:

\_label\_8:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

testSwitch:

#const define: sum

#load 3 to s1

add $s1, $0, $0

ori $s1, $s1, 3

sw $s1, ($sp)

addiu $sp, $sp, -4

#end const define: sum

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

#assign: i = 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

add $s3, $0, $s1

#write s3 to i

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: i = 0

#assign: int\_1 = 0

#load 0 to s1

add $s1, $0, $0

ori $s1, $s1, 0

add $s3, $0, $s1

#write s3 to int\_1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: int\_1 = 0

\_label\_20:

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

#load sum to s2

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s2, ($t0)

#load sum to s2

bge $s1, $s2, \_label\_19

#load int\_1 to s4

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s4, ($t0)

#load int\_1 to s4

#load 0 to s2

add $s2, $0, $0

ori $s2, $s2, 0

bne $s4, $s2, \_label\_22

#assign: char\_1 = 'a'

#load 'a' to s1

add $s1, $0, $0

ori $s1, $s1, 97

add $s3, $0, $s1

#write s3 to char\_1

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: char\_1 = 'a'

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load char\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

lw $s1, ($t0)

#load char\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

j \_label\_21

\_label\_22:

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

bne $s4, $s2, \_label\_23

#assign: char\_1 = 'b'

#load 'b' to s1

add $s1, $0, $0

ori $s1, $s1, 98

add $s3, $0, $s1

#write s3 to char\_1

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: char\_1 = 'b'

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load char\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

lw $s1, ($t0)

#load char\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

j \_label\_21

\_label\_23:

#assign: char\_1 = const\_char\_2

#load const\_char\_2 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 20

lw $s1, ($t0)

#load const\_char\_2 to s1

add $s3, $0, $s1

#write s3 to char\_1

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: char\_1 = const\_char\_2

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load char\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

lw $s1, ($t0)

#load char\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

j \_label\_21

\_label\_21:

#load char\_1 to s4

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

lw $s4, ($t0)

#load char\_1 to s4

#load 'a' to s2

add $s2, $0, $0

ori $s2, $s2, 97

bne $s4, $s2, \_label\_25

#add: @temp59 = i + 1

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

add $s3, $s1, $s2

#write s3 to @temp59

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp59 = i + 1

#assign: int\_1 = @temp59

#load @temp59 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp59 to s1

add $s3, $0, $s1

#write s3 to int\_1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: int\_1 = @temp59

#write int: int\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load int\_1 to s1

add $a0, $0, $s1

syscall

#end write int: int\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

j \_label\_24

\_label\_25:

#load 'b' to s2

add $s2, $0, $0

ori $s2, $s2, 98

bne $s4, $s2, \_label\_26

#assign: int\_1 = sum

#load sum to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load sum to s1

add $s3, $0, $s1

#write s3 to int\_1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: int\_1 = sum

#write int: int\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load int\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load int\_1 to s1

add $a0, $0, $s1

syscall

#end write int: int\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

j \_label\_24

\_label\_26:

\_label\_24:

#add: @temp60 = i + 1

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

#load 1 to s2

add $s2, $0, $0

ori $s2, $s2, 1

add $s3, $s1, $s2

#write s3 to @temp60

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp60 = i + 1

#assign: i = @temp60

#load @temp60 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp60 to s1

add $s3, $0, $s1

#write s3 to i

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: i = @temp60

j \_label\_20

\_label\_19:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

func\_ret\_int\_1:

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

#write string: "INPUT of func\_ret\_int\_1:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str6

syscall

#end write string: "INPUT of func\_ret\_int\_1:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: i\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load i\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load i\_1 to s1

add $a0, $0, $s1

syscall

#end write int: i\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: i\_2

add $v0, $0, $0

ori $v0, $v0, 1

#load i\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

add $t2, $s0, $t0

lw $s1, ($t2)

#load i\_2 to s1

add $a0, $0, $s1

syscall

#end write int: i\_2

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load c\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

add $t2, $s0, $t0

lw $s1, ($t2)

#load c\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load c\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load c\_2 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#assign: i\_temp = i\_1

#load i\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load i\_1 to s1

add $s3, $0, $s1

#write s3 to i\_temp

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: i\_temp = i\_1

#assign: i\_1 = i\_2

#load i\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

add $t2, $s0, $t0

lw $s1, ($t2)

#load i\_2 to s1

add $s3, $0, $s1

#write s3 to i\_1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

sw $s3, ($t2)

#end write s3

#end assign: i\_1 = i\_2

#assign: i\_2 = i\_temp

#load i\_temp to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i\_temp to s1

add $s3, $0, $s1

#write s3 to i\_2

add $t0, $0, $0

ori $t0, $t0, 12

add $t2, $s0, $t0

sw $s3, ($t2)

#end write s3

#end assign: i\_2 = i\_temp

#assign: c\_temp = c\_1

#load c\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

add $t2, $s0, $t0

lw $s1, ($t2)

#load c\_1 to s1

add $s3, $0, $s1

#write s3 to c\_temp

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: c\_temp = c\_1

#assign: c\_1 = c\_2

#load c\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load c\_2 to s1

add $s3, $0, $s1

#write s3 to c\_1

add $t0, $0, $0

ori $t0, $t0, 8

add $t2, $s0, $t0

sw $s3, ($t2)

#end write s3

#end assign: c\_1 = c\_2

#assign: c\_2 = c\_temp

#load c\_temp to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load c\_temp to s1

add $s3, $0, $s1

#write s3 to c\_2

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

sw $s3, ($t2)

#end write s3

#end assign: c\_2 = c\_temp

#write string: "OPERATE of func\_ret\_int\_1:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str7

syscall

#end write string: "OPERATE of func\_ret\_int\_1:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: i\_1

add $v0, $0, $0

ori $v0, $v0, 1

#load i\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load i\_1 to s1

add $a0, $0, $s1

syscall

#end write int: i\_1

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write int: i\_2

add $v0, $0, $0

ori $v0, $v0, 1

#load i\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 12

add $t2, $s0, $t0

lw $s1, ($t2)

#load i\_2 to s1

add $a0, $0, $s1

syscall

#end write int: i\_2

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load c\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 8

add $t2, $s0, $t0

lw $s1, ($t2)

#load c\_1 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#load c\_2 to s1

add $t0, $0, $0

ori $t0, $t0, 4

add $t2, $s0, $t0

lw $s1, ($t2)

#load c\_2 to s1

add $a0, $0, $s1

syscall

#end write char

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return: i\_1

#load i\_1 to s1

add $t0, $0, $0

ori $t0, $t0, 16

add $t2, $s0, $t0

lw $s1, ($t2)

#load i\_1 to s1

add $v0, $0, $s1

addi $sp, $s0, 16

jr $ra

#end return: i\_1

#return:

addi $sp, $s0, 16

jr $ra

#end return:

j main

testPara:

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

sw $0, ($sp)

addiu $sp, $sp, -4

#assign: i = const\_int\_1

#load const\_int\_1 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 0

lw $s1, ($t0)

#load const\_int\_1 to s1

add $s3, $0, $s1

#write s3 to i

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: i = const\_int\_1

#assign: j = const\_int\_4

#load const\_int\_4 to s1

addi $t0, $0, 0x10010000

addi $t0, $t0, 12

lw $s1, ($t0)

#load const\_int\_4 to s1

add $s3, $0, $s1

#write s3 to j

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: j = const\_int\_4

#assign: c = 'A'

#load 'A' to s1

add $s1, $0, $0

ori $s1, $s1, 65

add $s3, $0, $s1

#write s3 to c

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: c = 'A'

#assign: d = 'Z'

#load 'Z' to s1

add $s1, $0, $0

ori $s1, $s1, 90

add $s3, $0, $s1

#write s3 to d

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

sw $s3, ($t0)

#end write s3

#end assign: d = 'Z'

#push: i

#load i to s1

add $t0, $0, $0

ori $t0, $t0, 8

sub $t0, $s0, $t0

lw $s1, ($t0)

#load i to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: i

#push: j

#load j to s1

add $t0, $0, $0

ori $t0, $t0, 12

sub $t0, $s0, $t0

lw $s1, ($t0)

#load j to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: j

#push: c

#load c to s1

add $t0, $0, $0

ori $t0, $t0, 16

sub $t0, $s0, $t0

lw $s1, ($t0)

#load c to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: c

#push: d

#load d to s1

add $t0, $0, $0

ori $t0, $t0, 20

sub $t0, $s0, $t0

lw $s1, ($t0)

#load d to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: d

#call: func\_ret\_int\_1

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal func\_ret\_int\_1

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: func\_ret\_int\_1

#write int: @RET34

add $v0, $0, $0

ori $v0, $v0, 1

#load @RET34 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET34 to s1

add $a0, $0, $s1

syscall

#end write int: @RET34

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

j main

testRecursion:

#push: 2

#load 2 to s1

add $s1, $0, $0

ori $s1, $s1, 2

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 2

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#push: 3

#load 3 to s1

add $s1, $0, $0

ori $s1, $s1, 3

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 3

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#mul: @temp61 = @RET35 \* @RET36

#load @RET35 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET35 to s1

#load @RET36 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET36 to s2

mult $s1, $s2

mflo $s3

#write s3 to @temp61

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end mul: @temp61 = @RET35 \* @RET36

#push: @temp61

#load @temp61 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp61 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp61

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#push: 3

#load 3 to s1

add $s1, $0, $0

ori $s1, $s1, 3

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 3

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#push: 4

#load 4 to s1

add $s1, $0, $0

ori $s1, $s1, 4

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 4

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#add: @temp62 = @RET38 + @RET39

#load @RET38 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET38 to s1

#load @RET39 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET39 to s2

add $s3, $s1, $s2

#write s3 to @temp62

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp62 = @RET38 + @RET39

#push: @temp62

#load @temp62 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp62 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp62

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#sub: @temp63 = @RET37 - @RET40

#load @RET40 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET40 to s2

#load @RET37 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET37 to s1

sub $s3, $s1, $s2

#write s3 to @temp63

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end sub: @temp63 = @RET37 - @RET40

#push: 2

#load 2 to s1

add $s1, $0, $0

ori $s1, $s1, 2

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 2

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#push: 3

#load 3 to s1

add $s1, $0, $0

ori $s1, $s1, 3

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 3

#call: Fibonacci

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal Fibonacci

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: Fibonacci

#add: @temp64 = @RET41 + @RET42

#load @RET41 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @RET41 to s1

#load @RET42 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET42 to s2

add $s3, $s1, $s2

#write s3 to @temp64

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end add: @temp64 = @RET41 + @RET42

#push: @temp64

#load @temp64 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp64 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp64

#call: fac

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal fac

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: fac

#div: @temp65 = @temp63 / @RET43

#load @RET43 to s2

addiu $sp, $sp, 4

lw $s2, ($sp)

#load @RET43 to s2

#load @temp63 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp63 to s1

div $s1, $s2

mflo $s3

#write s3 to @temp65

sw $s3, ($sp)

addiu $sp, $sp, -4

#end write s3

#end div: @temp65 = @temp63 / @RET43

#push: @temp65

#load @temp65 to s1

addiu $sp, $sp, 4

lw $s1, ($sp)

#load @temp65 to s1

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: @temp65

#push: 'a'

#load 'a' to s1

add $s1, $0, $0

ori $s1, $s1, 97

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 'a'

#push: 'b'

#load 'b' to s1

add $s1, $0, $0

ori $s1, $s1, 98

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 'b'

#push: 'c'

#load 'c' to s1

add $s1, $0, $0

ori $s1, $s1, 99

sw $s1, ($sp)

addiu $sp, $sp, -4

#end push: 'c'

#call: hanoi

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal hanoi

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: hanoi

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

main:

add $s0, $0, $sp

la $ra, end

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str8

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "Start testing global:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str9

syscall

#end write string: "Start testing global:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#call: testGlobal

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testGlobal

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testGlobal

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str10

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "Start testing return:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str11

syscall

#end write string: "Start testing return:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#call: testReturn

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testReturn

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testReturn

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str12

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "Start testing I/O:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str13

syscall

#end write string: "Start testing I/O:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#call: testIO

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testIO

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testIO

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str14

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "Start testing assign & exp:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str15

syscall

#end write string: "Start testing assign & exp:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#call: testAssignAndExp

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testAssignAndExp

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testAssignAndExp

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str16

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "Start testing if & while:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str17

syscall

#end write string: "Start testing if & while:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#call: testIfWhile

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testIfWhile

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testIfWhile

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str18

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "Start testing switch:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str19

syscall

#end write string: "Start testing switch:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#call: testSwitch

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testSwitch

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testSwitch

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str20

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "Start testing parameter:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str21

syscall

#end write string: "Start testing parameter:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#call: testPara

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testPara

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testPara

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str22

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#write string: "Start testing recursion:"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str23

syscall

#end write string: "Start testing recursion:"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#call: testRecursion

sw $s0, ($sp)

addiu $sp, $sp, -4

sw $ra, ($sp)

addiu $sp, $sp, -4

addi $s0, $sp, 8

jal testRecursion

subi $t0, $s0, 4

lw $ra, ($t0)

lw $s0, ($s0)

sw $v0, ($sp)

addiu $sp, $sp, -4

#end call: testRecursion

#write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

add $v0, $0, $0

ori $v0, $v0, 4

la $a0, str24

syscall

#end write string: "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

#write char

add $v0, $0, $0

ori $v0, $v0, 11

#enter

add $s1, $0, $0

ori $s1, $s1, 10

add $a0, $0, $s1

syscall

#end write char

#return:

addi $sp, $s0, 0

jr $ra

#end return:

#return:

addi $sp, $s0, 0

jr $ra

#end return:

end:

测试输入:

1

a

1

a

a1

测试输出:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Start testing global:

12345679

0

0

-12345679

9

\_

+

\*

12345679

-12345679

9

\*

0

1

1

2

3

1

1

2

6

2

a

A

z

Z

\_

+

-

\*

/

6

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Start testing return:

22

a

b

c

\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Start testing I/O:

1

a!@

#$^&\*()Qqaa123[];',./

1

a

1

a!@

#$^&\*()Qqaa123[];',./1

a

a1

!@

#$^&\*()Qqaa123[];',./a

1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Start testing assign & exp:

1

40320

-2

1

3

a

\_

a

\_

903264

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Start testing if & while:

+

-

+

-

+

/

\*

/

\*

/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Start testing switch:

a

1

b

3

\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Start testing parameter:

INPUT of func\_ret\_int\_1:

12345679

-12345679

A

Z

OPERATE of func\_ret\_int\_1:

-12345679

12345679

Z

A

-12345679

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Start testing recursion:

take

1

from

a

to

b

take

2

from

a

to

c

take

1

from

b

to

c

take

3

from

a

to

b

take

1

from

c

to

a

take

2

from

c

to

b

take

1

from

a

to

b

take

4

from

a

to

c

take

1

from

b

to

c

take

2

from

b

to

a

take

1

from

c

to

a

take

3

from

b

to

c

take

1

from

a

to

b

take

2

from

a

to

c

take

1

from

b

to

c

测试程序参考宋卓洋同学在讨论区的优秀贡献