```
#############################################################################
# Title: Assign02P3                   Author: Noah del Angel
# Class: CS 23182138.002, Fall 2020       Submitted: 11/5/2020
#############################################################################
# Program: MIPS tranlation of a given C++ program
#############################################################################
# Pseudocode description: supplied a2p2_SampSoln.cpp
#############################################################################
             .data
a1:          .space 48
a2:          .space 48
a3:          .space 48
einStr:      .asciiz "\nEnter integer #"
moStr:       .asciiz "Max of "
ieStr:       .asciiz " ints entered..."
emiStr:      .asciiz "Enter more ints? (n or N = no, others = yes) "
begA1Str:    .asciiz "\nbeginning a1: "
nn09A1Str:   .asciiz "a1 (noneg09): "
procA1Str:   .asciiz "processed a1: "
procA2Str:   .asciiz "          a2: "
procA3Str:   .asciiz "          a3: "
dacStr:      .asciiz "Do another case? (n or N = no, others = yes) "
dlStr:       .asciiz "\n================================="
byeStr:      .asciiz "bye.."

             .text
             .globl main
main:

             li $t8, # $t8 has reply

             j WTest1

             begW1:
             li $t1, # $t1 has used1
             la $t4, # $t4 has hopPtr1

             j W2Test

             begW2:
             # cout einStr
             li $v0, 4
             la $a0, einStr
             syscall

             # cout used1 + 1
             addi $v1, $t1, 1 # $t1 has used + 1
```

```
li $v0, 1
move $a0, $v1
syscall

# cout ':' and ' '
li $v0, 11
li $a0, ':'
syscall

li $a0, ' '
syscall

#cin *hopPtr1
li $v0, 5
syscall

sw $v0, 0($t4)

addi $t4, $t4, 4 # *hopPtr1++
addi $t1, $t1, 1 # used1++

# if used1 => 12
li $v0, 12
bge $t1, $v0, else1

 # cout emiStr
 li $v0, 4
 la $a0, emiStr
 syscall

 # cin reply
 li $v0, 12
 syscall

 move $t

 j endI1

else1:
# cout moStr and 12 and ieStr
li $v0, 4
la $a0, moStr
syscall

li $v0, 1
li $a0, 12
```

```
syscall

li $v0, 4
la $a0, ieStr
syscall

li $v0, 11
li $a0, '\n'
syscall

li $t8, 'n' # reply = 'n'
endI1:
endW2:
W2Test:

li $a0, 'n'
beq $a0, $t8, xitW2 # check if $t8 == n

li $a0, 'N'
bne $a0, $t8, begW2 # check if $t8 != N
xitW2:

#cout begA1sStr
li $v0, 4
la $a0, begA1Str
syscall

# if used1 <= 0
blez $t1, endI2

la $t4, a1 # hopPtr1 = a1


# endPtr1 = a1 + used1
sll $v1, $t1, 2
add $a1, $t4, $v1

begDW1:
# Cout *hopPtr1 and ' ' and ' '
li $v0, 1
lw $a0, 0($t4)
syscall

li $v0, 11
li $a0, ' '
syscall
syscall
```

```
addi $t4, $t4, 4 # ++hopPtr1
endDW1:
DWTest1:
# if hopPtr1 < endPtr1
blt $t4, $a1, begDW1

endI2:

li $v0, 11
li $a0, '\n'
syscall

# if used1 <= 0
blez $t1, endI3
la $t4, a1 # hopPtr1 = a1

# endPtr1 = a1 + used1
sll $v1, $t1, 2
add $a1, $t4, $v0

j FTest1
begF1:

lw $t0, 0($t4)  # target = *hopPtr1

# if target < 0 && target <= 9
bltz $t0, begI4
li $v0, 9
ble $t0, $v0, endI4

begI4:


# hopPtr11 = hopPtr1 + 1
la $t6, a1
addi $t6, $t4, 4

j FTest2
begF2:
#  *(hopPtr11 - 1) = *hopPtr11
lw $v0, 0($t6)
sw $v0, -4($t6)

#  ++hopPtr11
addi $t6, $t6, 4
endF2:
```

```
FTest2:
blt $t6, $a1, begF2 # if hopPtr11 < endPtr1


addi $t1, $t1, -1 # used 1 --
addi $a1, $a1, -4 # endPtr1 --
addi $t4, $t4, -4 # hopPtr 1 --
endI4:
addi $t4, $t4, 4 # hopPtr 1 ++

endF1:
FTest1:
blt $t4, $a1, begF1 # if hopPtr1 < endPtr1


# cout nn09A1Str
li $v0, 4
la $a0, nn09A1Str
syscall

# if used1 <= 0
blez $t1, endI5

la $t4, a1 # hopPtr1 = a1

# endPtr1 = a1 + used1
sll $v1, $t1, 2
add $a1, $t4, $v1

begDW2:
# cout *hopPtr ' ' ' '
li $v0, 1
lw $a0, 0($t4)
syscall


li $v0, 11
li $a0, ' '
syscall
syscall

# ++hopPtr1
addi $t4, $t4, 4
endDW2:
DWTest2:
blt $t4, $a1, begDW2 # if hopPtr1 < endPtr1
```

```
endI5:

li $v0, 11
li $a0, '\n'
syscall


li $t2, 0 # used2 = 0
li $t3, 0 # used3 = 0

la $t4, a1 # hopPtr1 = a1
la $t5, a2 # hopPtr2 = a2
la $t7, a3 # hopPtr3 = a3

        # endPtr1 = a1 + used1
        sll $v0, $t1, 2
        add $a1, $t4, $v0

 # goto WTest3
 j WTest3
 begW3:
        # intHolder = *hopPtr1
        lw $t8, 0($t4)
        # *hopPtr2 = intHolder;
        sw $t8, 0($t5)

        # ++used2
        addi $t2, $t2, 1

        # ++hopPtr2
        addi $t5, $t5, 4

        # *hopPtr3 = intHolder
        sw $t8, 0($t7)

        # ++used3
        addi $t3, $t3, 1

        # ++hopPtr3
        addi $t7, $t7, 4

# ++ hopPtr1
        addi $t4, $t4, 4
 endW3:
 WTest3:
        blt $t4, $a1, begW3
 # iter = 0
 li $t8, 0
 begDW3:

 # ++iter
 addi $t8, $t8, 1

 # count = 0
 li $9, 0

 # if ( iter != 1 )
 li $v0, 1
 bne $t8, $v0, elseI6
        # hopPtr1 = a1
        la $t4, a1

        #endPtr1 = a1 + used1
        sll $v0, $t1, 2
        add $a1, $t4, $v0

        j FTest3
        begF3:
        # target = *hopPtr1
        lw $t0, 0($t4)

        # if target == 5
li $v1, 5
beq $t0, $v1, elseI7
# ++count
addi $t9, $t9, 1

# goto endI7
j endI7

elseI7:
# if count == 0
beqz $t9, endI8
#  *(hopPtr1 - count) = *hopPtr1
lw $v0, 0($t4)
sll $a0, $t9, 2
sub $t4, $t4, $a0
sw $v0, 0($t4)
add $t4, $t4, $a0

endI8:
endI7:
# ++hopPtr1
```

```
addi $t4, $t4, 4

        endF3:
        FTest3:
        # if hopPtr1 < endPtr1
        blt $t4, $a1, begF3

        # used -= count
        sub $t1, $t1, $t9

# if used1 != 0
bnez $t1, endI9
# hopPtr1 = a1
        la $t4, a1

        li $a0, -99
        sw $a0, 0($t4) # *hopPtr1 = -99

        addi $t1, $t1, 1 # used1++

 endI9:
 j endI6

 elseI6:
        li $v1, 2

        # if iter != 2
        bne $t8, $v1, else10
        # hopPtr = a2
        la $t5, a2

# endPtr2 = a2 + used2
sll $v1, $t2, 2
add $a2, $t5, $v1

j FTest4

begF4:
lw $t0, 0($t5) # target = *hopPtr2

li $v1, 4

ble $t0, $v1, elseI11
addi $t9, $t9, 1 # ++ count
j endI11
elseI11:
beqz $t9, endI12
```

```
#*(hopPtr2 - count) = *hopPtr2
lw $v0, 0($t5)
sll $a0, $t9, 2
sub $t5, $t5, $a0
sw $v0, 0($t5)
add $t5, $t5, $a0

endI12:
 endI11:
 # ++hopPtr2
 addi $t5, $t5, 4
endF4:
FTest4:
# if hopPtr < endPtr2
blt $t5, $a2, begF4

# used 2 -= count
sub $t4, $t4, $t9

# if used2 != 0
bnez $t2, endI13
la $t5, a2 # hopPtr2 = a2

li $a0, -99
sw $a0, 0($t5) # *hopPtr2 = -99
addi $t2, $t2, 1 # ++used2

endI13:
# goto endI10
j endI10
else10:
# hopPtr3 = a3
la $t7, a3

# endPtr3 = a3 + used3
sll $v1, $t3, 2
add $a3, $t7, $v1

j Ftest5
begF5:
lw $t0, # target = *hopPtr3

# if target >= 6
li $a1, 6
bge $t0, $a1, else14
addi $t9, $t9, 1 # count ++
j endI14
else14:
# if ( count == 0 )
beqz $t9, endI15
#*(hopPtr3 - count) = *hopPtr3
lw $v0, 0($t7)
sll $a0, $t9, 2
sub $t5, $t7, $a0
sw $v0, 0($t7)
add $t7, $t7, $a0
endI15:
endI14:
addi $t7, $t7, 4 # ++hopPtr3

endF5:
Ftest5:
# if hopPtr3 < endPtr3
blt $t7, $a3, begF5

# used3 -= count
sub $t3, $t9, $t9

# if used3 != 0
bnez $t3, endI16
la $t7 , a3 # hopPtr3 = a3
li $a0, -99
sw $a0, 0($t7) # *hopPtr3 = -99

addi $t3, $t3, 1 # ++used
endI16:
endI10:
 endI6:

endDW3:

DWTest3:
# if iter < 3
li $a0, 3
blt $t8, $a0, begDW3
endI3:
# cout << procA1Str
li $v0, 4
la $a0, procA1Str
syscall

# if used1 <= 0
blez $t1, endI17
```

```
la $t4, a1 # hopPtr1 = a1
# endPtr1 = a1 + used1
sll $v1, $t1, 2
add $a1, $t4, $v1

begDW4:
# cout *hopPtr1 ' ' and ' '
li $v0, 1
lw $a0, 0($t4)
syscall

li $v0, 11
li $a0, ' '
syscall
syscall

# ++hopPtr1
addi $t4, $t4, 4

endDW4:
DWTest4:
blt $t4, $a1, begDW4
endI17:

li $v0, 11
li $a0, '\n'
syscall

li $v0, 4
la $a0, procA2Str
syscall

blez $t2, endI18
# hopPtr2 = a2
la $t5, a2

# endPtr2 = a2 + used2
sll $v1, $t2, 2
add $a2, $t5, $v1

begDW5:
# cout *hopPtr2 ' ' ' '
li $v0, 1
lw $a0, 0($t5)
syscall

li $v0, 11
li $a0, ' '
syscall
syscall

addi $t5, $t5, 4 # hopPtr2++

DWTest5:
blt $t5, $a2, begDW5
endI18:

li $v0, 11
li $a0, ' '
syscall


# cout procA3Str
li $v0, 4
la $a0, procA3Str
syscall

# if used3 <= 0
blez $t3, endI19
# endPtr3 = a3
la $t7, a3

# endPtr3 = a3 + used3
sll $v1, $t3, 2
add $a3, $t7, $v1

begDW6:
# cout *hoptr3 ' ' and ' '
li $v0, 1
lw $a0, 0($t7)
syscall

li $v0, 11
li $a0, ' '
syscall
syscall

# ++hopPtr3
addi $t7, $t7, 4
endDw6:
DWTest6:
# if (hopPtr3 < endPtr3)
blt $t7, $a3, begDW6
```

```
endI19:

li $v0, 11
li $a0, ' '
syscall

# cout dacStr
li $v0, 4
la $a0, dacStr
syscall

# cin reply
li $v0, 12
        syscall

        move $t8, $v0 #$t8 has reply


WTest1:
li $a0, # $a0 has n
beq $a0,# check if $t8 != n

li $a0, # $a0 has N
bne  $a(# check if $t8 != N

xitW1:
li $v0, 4
la $a0, dlStr
syscall

la $a0, byeStr
syscall

la $a0, dlStr
syscall

#Graceful exit
li $v0, 10
syscall
```

###

###

###

###