```
###############################################################################
# Title: Assign02P3                   Author: Ethan West
# Class: CS 2318-25?, Spring 2024     Submitted: 04/18/2024
###############################################################################
# Program: MIPS tranlation of a given C++ program
###############################################################################
# Pseudocode description: supplied a2p2_SampSoln.cpp
###############################################################################

##include <iostream>
#using namespace std;

#int a1[12], a2[12], a3[12], a4[12];
#int used1, used2, used3, used4, minInt, intNum, oneInt;
#int* hopPtr;
#int* hopPtr1;
#int* hopPtr2;
#int* hopPtr3;
#int* hopPtr4;
#int* endPtr;
#int* endPtr1;
#int* endPtr2;
#int* iPtr;
#char reply;
#char begA1Str[] = "beginning a1: ";
#char cpaA1Str[] = "chkPointA a1: ";
#char proA1Str[] = "processed a1: ";
#char comAeStr[] = "          a";
#char comAfStr[] = ": ";
#char einStr[]   = "Enter integer #";
#char moStr[]    = "Max of ";
#char ieStr[]    = " ints entered...";
#char eaiStr[]   = "End adding ints? (y or Y = yes, others = no) ";
#char dacStr[]   = "Do another case? (n or N = no, others = yes) ";
#char dlStr[]    = "================================";
#char byeStr[]   = "bye...";

            .data
a1:         .space 40
a2:         .space 40
a3:         .space 40
a4:         .space 40
begA1Str:   .asciiz "\nbeginning a1: "
cpaA1Str:   .asciiz "chkPointA a1: "
proA1Str:   .asciiz "processed a1: "
comAeStr:   .asciiz "     a"
comAfStr:   .asciiz ": "
einStr:     .asciiz "\nEnter integer #"
moStr:      .asciiz "Max of "
ieStr:      .asciiz " ints entered..."
eaiStr:     .asciiz "End adding ints? (y or Y = yes, others = no) "
```

```
dacStr:     .asciiz "Do another case? (n or N = no, others = yes) "
dlStr:      .asciiz "\n================================"
byeStr:     .asciiz "\nbye..."

#int main()
#{
            .text
            .globl main
main:

################################################
# Register usage:
#################
# $a0: short-lived holder 3
# $a1: used1
# $a2: used2
# $a3: used3
# $v1: used4
# $t0: short-lived holder 1
# $t1: hopPtr1
# $t2: hopPtr2
# $t3: hopPtr3 or hopPtr
# $t4: hopPtr4 or endPtr
# $t5: intNum or iPtr
# $t6: minInt or reply
# $t7: oneInt
# $t8: endPtr2
# $t9: endPtr1
# $v0: short-lived holder 2
################################################

#           //do
begDW1:#    {
#               intNum = 0;
                li $t5, 0
#               used1 = 0;
                li $a1, 0
#               used2 = 0;
                li $a2, 0
#               hopPtr1 = a1;
                la $t1, a1
#               hopPtr2 = a2;
                la $t2, a2
#               cout << eaiStr;
                li $v0, 4
                la $a0, eaiStr
                syscall
#               cin >> reply;
                li $v0, 12
                syscall
                move $t6, $v0
```

```

#                  //while (reply != 'y' && reply != 'Y')
#               goto WTest1;
               j WTest1
begW1:#            {
#                  ++intNum;
                   addi $t5, $t5, 1
#                   cout << einStr;
                   li $v0, 4
                   la $a0, einStr
                   syscall
#                   cout << intNum;
                   li $v0, 1
                   move $a0, $t5
                   syscall
#                   cout << ':' << ' ';
                   li $v0, 4
                   la $a0, comAfStr
                   syscall
#                   cin >> oneInt;
                   li $v0, 5
                   syscall
                   move $t7, $v0
#                      //if ( (intNum & 1) != 0 )
#                   if ( (intNum & 1) == 0 ) goto else1;
                   andi $t0, $t5, 1
                   beqz $t0, else1
begI1:#            {
#                      *hopPtr1 = oneInt;
                       sw $t7, 0($t1)
#                      ++hopPtr1;
                       addi $t1, $t1, 4
#                      ++used1;
                       addi $a1, $a1, 1
#                      goto endI1;
                       j endI1
#//                    }
else1:#//              else
#//                    {
#                          *hopPtr2 = oneInt;
                           sw $t7, 0($t2)
#                          ++hopPtr2;
                           addi $t2, $t2, 4
#                          ++used2;
                           addi $a2, $a2, 1
endI1:#//                  }
#                      //if (intNum == 12)
#                      if (intNum != 12) goto else2;
                       li $t0, 12
                       bne $t5, $t0, else2
begI2:#//               {
```

```
#                          cout << moStr;
                           li $v0, 4
                           la $a0, moStr
                           syscall
#                          cout << 12;
                           li $v0, 1
                           li $a0, 12
                           syscall
#                          cout << ieStr;
                           li $v0, 4
                           la $a0, ieStr
                           syscall
#                          cout << endl;
                           li $v0, 11
                           li $a0, '\n'
                           syscall
#                          reply = 'y';
                           li $t6, 'y'
#                              goto endI2;
                           j endI2
#//                    }
else2:#//              else
#//                    {
#                          cout << eaiStr;
                           li $v0, 4
                           la $a0, eaiStr
                           syscall
#                          cin >> reply;
                           li $v0, 12
                           syscall
                           move $t6, $v0
endI2:#//              }
endW1:#//              }
WTest1:#
#                          if (reply == 'y') goto xitW1;
#                          if (reply != 'Y') goto begW1;
                       li $t0, 'y'
                       beq $t6, $t0, xitW1
                       li $t0, 'Y'
                       bne $t6, $t0, begW1
xitW1:#
#                      cout << endl;
                       li $v0, 11
                       li $a0, '\n'
                       syscall
#                      cout << begA1Str;
                       li $v0, 4
                       la $a0, begA1Str
                       syscall
#                      hopPtr = a1;
                       la $t3, a1
```

```
#                      endPtr = hopPtr + used1;
                       sll $t0, $a1, 2
                       add $t4, $t3, $t0
#                      //while (hopPtr < endPtr)
#                      goto WTest2;
                       j WTest2
begW2:#//                  {
#                          cout << *hopPtr << ' ' << ' ';
                           li $v0, 1
                           lw $a0, 0($t3)
                           syscall
                           li $v0, 11
                           li $a0, ' '
                           syscall
                           li $v0, 11
                           li $a0, ' '
                           syscall
#                          ++hopPtr;
                           addi $t3, $t3, 4
endW2:#//              }
WTest2:#               if (hopPtr < endPtr) goto begW2
                       blt $t3, $t4, begW2
#
#                      cout << endl;
                       li $v0, 11
                       li $a0, '\n'
                       syscall
#
#                      cout << comAeStr << 2 << comAfStr;
                       li $v0, 4
                       la $a0, comAeStr
                       syscall
                       li $v0, 1
                       li $a0, 2
                       syscall
                       li $v0, 4
                       la $a0, comAfStr
                       syscall
#                      hopPtr = a2;
                       la $t3, a2
#                      endPtr = hopPtr + used2;
                       sll $t0, $a2, 2
                       add $t4, $t3, $t0
#                      //while (hopPtr < endPtr)
#                      goto WTest3;
                       j WTest3
begW3:#//                  {
#                          cout << *hopPtr << ' ' << ' ';
                           li $v0, 1
                           lw $a0, 0($t3)
                           syscall
                           li $v0, 11
                           li $a0, ' '
                           syscall
                           li $v0, 11
                           li $a0, ' '
                           syscall
#                          ++hopPtr;
                           addi $t3, $t3, 4
endW3:#//                  }
WTest3:#                   if (hopPtr < endPtr) goto begW3;
                       blt $t3, $t4, begW3
#                          cout << endl;
                       li $v0, 11
                       li $a0, '\n'
                       syscall
#                      //if (used1 > 0 || used2 > 0)
#                      if (used1 > 0) goto begI3;
                       bgtz $a1, begI3
#                      if (used2 <= 0) goto else3;
                       blez $a2, else3
begI3:#//              {
#                          hopPtr1 = a1;
                           la $t1, a1
#                          hopPtr2 = a2;
                           la $t2, a2
#                              hopPtr3 = a3;
                           la $t3, a3
#                              hopPtr4 = a4;
                           la $t4, a4
#                          endPtr1 = hopPtr1 + used1;
                           sll $t0, $a1, 2
                           add $t9, $t1, $t0
#                          endPtr2 = hopPtr2 + used2;
                           sll $t0, $a2, 2
                           add $t8, $t2, $t0
#                          used3 = 0;
                           li $a3, 0
#                          used4 = 0;
                           li $v1, 0
#                          //if (used1 > 0)
#                          if (used1 <= 0) goto else4;
                           blez $a1, else4
begI4:#//                  {
#                              minInt = *hopPtr1;
                               lw $t6, 0($t1)
#                              goto endI4;
                               j endI4
#//                        }
else4:#//                      else
#//                            {
#                              minInt = *hopPtr2;
```

```
                               lw $t6, 0($t2)
endI4:#//                      }
#                              //while (hopPtr1 < endPtr1 && hopPtr2 < endPtr2)
#                              goto WTest4;
                           j WTest4
begW4:#//                      {
#                              //while (hopPtr1 < endPtr1)
#                                  goto WTest5;
                               j WTest5
begW5:#//                      {
#                               oneInt = *hopPtr1;
                               lw $t7, 0($t1)
#                               //if (oneInt < minInt)
#                              if (oneInt >= minInt) goto endI5;
                               bge $t7, $t6, endI5
begI5:#//                       {
#                                      minInt = oneInt;
                                   move $t6, $t7
endI5:#//                       }
#                               //if ( (oneInt & 1) == 0 ) break;
#                               if ( (oneInt & 1) == 0 ) goto brk6;
                               andi $t0, $t7, 1
                               beqz $t0, brk6
#                               *hopPtr3 = oneInt;
                               sw $t7, 0($t3)
#                               ++used3;
                               addi $a3, $a3, 1
#                               ++hopPtr1;
                               addi $t1, $t1, 4
#                               ++hopPtr3;
                               addi $t3, $t3, 4
endW5:#//                      }
WTest5:#                       if (hopPtr1 < endPtr1) goto begW5;
                               blt $t1, $t9, begW5
brk6:#
#                              //while (hopPtr2 < endPtr2)
#                              goto WTest6;
                               j WTest6
begW6:#//                      {
#                                  oneInt = *hopPtr2;
                                   lw $t7, 0($t2)
#                                  //if (oneInt < minInt)
#                                  if (oneInt >= minInt) goto endI7;
                                   bge $t7, $t6, endI7
begI7:#//                          {
#                                      minInt = oneInt;
                                       move $t6, $t7
endI7:#//                          }
#                                  //if ( (oneInt & 1) != 0 ) break;
#                                  if ( (oneInt & 1) != 0 ) goto brk8;
                                   andi $t0, $t7, 1
                                   bnez $t0, brk8
#                                  *hopPtr4 = oneInt;
                                   sw $t7, 0($t4)
#                                  ++used4;
                                   addi $v1, $v1,  1
#                                  ++hopPtr2;
                                   addi $t2, $t2, 4
#                                  ++hopPtr4;
                                   addi $t4, $t4, 4
endW6:#//                      }
WTest6:#                       if (hopPtr2 < endPtr2) goto begW6;
                               blt $t2, $t8, begW6
brk8:#
#                              //if (hopPtr1 < endPtr1 && hopPtr2 < endPtr2)
#                              if (hopPtr1 >= endPtr1) goto endI9;
                               bge $t1, $t9, endI9
#                              if (hopPtr2 >= endPtr2) goto endI9;
                               bge $t2, $t8, endI9
begI9:#//                      {
#                                  *hopPtr3 = *hopPtr2;
                                   lw $t0, 0($t2)
                                   sw $t0, 0($t3)
#                                  *hopPtr4 = *hopPtr1;
                                   lw $t0, 0($t1)
                                   sw $t0, 0($t4)
#                                  ++used3;
                                   addi $a3, $a3, 1
#                                  ++used4;
                                   addi $v1, $v1, 1
#                                  ++hopPtr1;
                                   addi $t1, $t1, 4
#                                  ++hopPtr2;
                                   addi $t2, $t2, 4
#                                  ++hopPtr3;
                                   addi $t3, $t3, 4
#                                  ++hopPtr4;
                                   addi $t4, $t4, 4
endI9:#//                      }
endW4:#//                  }
WTest4:#                   if (hopPtr1 >= endPtr1) goto xitW4;
                           bge $t1, $t9, xitW4
#                          if (hopPtr2 < endPtr2) goto begW4;
                           blt $t2, $t8, begW4
xitW4:#
#
#                              //while (hopPtr1 < endPtr1)
#                          goto WTest7;
                           j WTest7
begW7:#//                      {
#                              oneInt = *hopPtr1;
                               lw $t7, 0($t1)
```

```
#                              //if (oneInt < minInt)
#                              if (oneInt >= minInt) goto endI10;
                               bge $t7, $t6, endI10
begI10:#//                         {
#                                      minInt = oneInt;
                                   move $t6, $t7
endI10:#//                     }
#                              //if ( (oneInt & 1) != 0 )
#                              if ( (oneInt & 1) == 0 ) goto else11;
                               andi $t0, $t7, 1
                               beqz $t0, else11
begI11:#//                         {
#                                  *hopPtr3 = oneInt;
                                   sw $t7, 0($t3)
#                                  ++used3;
                                   addi $a3, $a3, 1
#                                  ++hopPtr3;
                                   addi $t3, $t3, 4
#                                      goto endI11;
                                   j endI11
#//                            }
else11:#//                         else
#//                            {
#                                  *hopPtr4 = oneInt;
                                   sw $t7, 0($t4)
#                                  ++used4;
                                   addi $v1, $v1, 1
#                                  ++hopPtr4;
                                   addi $t4, $t4, 4
endI11:#//                     }
#                                      ++hopPtr1;
                                   addi $t1, $t1, 4
endW7:#//                  }
WTest7:#                   if (hopPtr1 < endPtr1) goto begW7;
                           blt $t1, $t9, begW7
#
#                          //while (hopPtr2 < endPtr2)
#                              goto WTest8;
                           j WTest8
begW8:#//                  {
#                                  oneInt = *hopPtr2;
                               lw $t7, 0($t2)
#                                  //if (oneInt < minInt)
#                                  if (oneInt >= minInt) goto endI12;
                               bge $t7, $t6, endI12
begI12:#//                         {
#                                  minInt = oneInt;
                                   move $t6, $t7
endI12:#//                         }
#                                  //if ( (oneInt & 1) != 0 )
#                                  if ( (oneInt & 1) == 0 ) goto else13;
```

```
                                andi $t0, $t7, 1
                                beqz $t0, else13
begI13:#//                          {
#                                   *hopPtr3 = oneInt;
                                    sw $t7, 0($t3)
#                                   ++used3;
                                    addi $a3, $a3, 1
#                                   ++hopPtr3;
                                    addi $t3, $t3, 4
#                                       goto endI13;
                                    j endI13
#//                                 }
else13:#//                          else
#//                                 {
#                                   *hopPtr4 = oneInt;
                                    sw $t7, 0($t4)
#                                   ++used4;
                                    addi $v1, $v1, 1
#                                   ++hopPtr4;
                                    addi $t4, $t4, 4
endI13:#//                      }
#                                   ++hopPtr2;
                                addi $t2, $t2, 4
endW8:#//                   }
WTest8:#                        if (hopPtr2 < endPtr2) goto begW8;
                            blt $t2, $t8, begW8
#                           goto endI3;
                            j endI3

#//                     }
else3:#//               else
#//                     {
#                           used3 = 0;
                        li $a3, 0
#                       used4 = 0;
                        li $v1, 0
endI3:#//               }
#                       cout << comAeStr << 3 << comAfStr;
                        li $v0, 4
                        la $a0, comAeStr
                        syscall
                        li $v0, 1
                        li $a0, 3
                        syscall
                        li $v0, 4
                        la $a0, comAfStr
                        syscall

#                       hopPtr = a3;
                        la $t3, a3
#                       endPtr = hopPtr + used3;
```

```
                       sll $t0, $a3, 2
                       add $t4, $t0, $t3
#                      //while (hopPtr < endPtr)
#                      goto WTest9;
                       j WTest9
begW9:#//              {
#                              cout << *hopPtr << ' ' << ' ';
                           li $v0, 1
                           lw $a0, 0($t3)
                           syscall
                           li $v0, 11
                           li $a0, ' '
                           syscall
                           li $v0, 11
                           li $a0, ' '
                           syscall

#                              ++hopPtr;
                           addi $t3, $t3, 4
endW9:#//              }
WTest9:#               if (hopPtr < endPtr) goto begW9;
                       blt $t3, $t4, begW9
#                      cout << endl;
                       li $v0, 11
                       li $a0, '\n'
                       syscall
#                      cout << comAeStr << 4 << comAfStr;
                       li $v0, 4
                       la $a0, comAeStr
                       syscall
                       li $v0, 1
                       li $a0, 4
                       syscall
                       li $v0, 4
                       la $a0, comAfStr
                       syscall

#                      hopPtr = a4;
                       la $t3, a4
#                      endPtr = hopPtr + used4;
                       sll $t0, $v1, 2
                       add $t4, $t0, $t3
#                      //while (hopPtr < endPtr)
#                      goto WTest10;
                       j WTest10
begW10:#//             {
#                          cout << *hopPtr << ' ' << ' ';
                           li $v0, 1
                           lw $a0, 0($t3)
                           syscall
                           li $v0, 11
                           li $a0, ' '
                           syscall
                           li $v0, 11
                           li $a0, ' '
                           syscall
#                          ++hopPtr;
                           addi $t3, $t3, 4
endW10:#//             }
WTest10:#              if (hopPtr < endPtr) goto begW10;
                       blt $t3, $t4, begW10
#                      cout << endl;
                       li $v0, 11
                       li $a0, '\n'
                       syscall
#                      //if (used1 > 0 || used2 > 0)
#                      if (used1 > 0) goto begI14;
                       bgtz $a1, begI14
#                      if (used2 <= 0) goto endI14;
                       blez $a2, endI14
begI14:#//             {
#                              used1 = 0;
                           li $a1, 0
#                              used2 = 0;
                           li $a2, 0
#                              hopPtr = a3;
                           la $t3, a3
#                              endPtr = hopPtr + used3;
                           sll $t0, $a3, 2
                           add $t4, $t0, $t3
#                              //while (hopPtr < endPtr)
#                          goto WTest11;
                           j WTest11
begW11:#//                 {
#                              oneInt = *hopPtr;
                               lw $t7, 0($t3)
#                              //for (iPtr = a1 + used1; iPtr > a1; --iPtr)
#                                  iPtr = a1 + used1;
                               la $a0, a1
                               sll $t0, $a1, 2
                               add $t5, $t0, $a0
#                                  goto FTest1;
                               j FTest1
begF1:#//                      {
#                                  //if ( *(iPtr - 1) <= oneInt ) break;
#                                  if ( *(iPtr - 1) <= oneInt ) goto brk15;
                                   lw $t0, -4($t5)
                                   ble $t0, $t7, brk15
#                                  *iPtr = *(iPtr - 1);
                                   lw $t0, -4($t5)
                                   sw $t0, 0($t5)
#                                  --iPtr;
```

```
                                   addi $t5, $t5, -4
endF1:#//                      }
FTest1:#                       if (iPtr > a1) goto begF1;
                               la $t0, a1
                               bgt $t5, $t0, begF1
brk15:#
#                              *iPtr = *hopPtr;
                               lw $t0, 0($t3)
                               sw $t0, 0($t5)
#                              ++used1;
                               addi $a1, $a1, 1
#                              ++hopPtr;
                               addi $t3, $t3, 4
endW11:#//                     }
WTest11:#                  if (hopPtr < endPtr) goto begW11;
                           blt $t3, $t4, begW11
#
#                              hopPtr = a4;
                           la $t3, a4
#                          endPtr = hopPtr + used4;
                           sll $t0, $v1, 2
                           add $t4, $t0, $t3
#                          //while (hopPtr < endPtr)
#                          goto WTest12;
                           j WTest12
begW12:#//                 {
#                              oneInt = *hopPtr;
                               lw $t7, 0($t3)
#                              //for (iPtr = a2 + used2; iPtr > a2; --iPtr)
#                                  iPtr = a2 + used2;
                               la $t5, a2
                               sll $t0, $a2, 2
                               add $t5, $t5, $t0
#                              goto FTest2;
                               j FTest2
begF2:#//                      {
#                                  //if ( *(iPtr - 1) <= oneInt ) break;
#                                  if ( *(iPtr - 1) <= oneInt ) goto brk16;
                                   lw $t0, -4($t5)
                                   ble $t0, $t7, brk16
#                                  *iPtr = *(iPtr - 1);
                                   lw $t0, -4($t5)
                                   sw $t0, 0($t5)
#                                  --iPtr;
                                   addi $t5, $t5, -4
endF2:#//                          }
FTest2:#                           if (iPtr > a2) goto begF2;
                               la $t0, a2
                               bgt $t5, $t0, begF2
brk16:#
#
```

```
#                              *iPtr = *hopPtr;
                               lw $t0, 0($t3)
                               sw $t0, 0($t5)
#                              ++used2;
                               addi $a2, $a2, 1
#                              ++hopPtr;
                               addi $t3, $t3, 4
endW12:#//                     }
WTest12:#                  if (hopPtr < endPtr) goto begW12;
                           blt $t3, $t4, begW12
#                              cout << cpaA1Str;
                           li $v0, 4
                           la $a0, cpaA1Str
                           syscall
#                              hopPtr = a1;
                           la $t3, a1
#                              endPtr = hopPtr + used1;
                           sll $t0, $a1, 2
                           add $t4, $t0, $t3
#

#                              //while (0 == 0)
#                              goto WTest13;
                           j WTest13
begW13:#//                     {
#                                  //if (hopPtr == a4 + used4 && endPtr == a4 + used4) break;
#                                  ////if (hopPtr == a4 + used4 && endPtr == a4 + used4) goto brk17;
#                                  if (hopPtr != a4 + used4) goto nbk17;
                               la $a0, a4
                               sll $t0, $v1, 2
                               add $t0, $t0, $a0
                               bne $t3, $t0, nbk17
#                                  if (endPtr == a4 + used4) goto brk17;
                               la $a0, a4
                               sll $t0, $v1, 2
                               add $t0, $t0, $a0
                               beq $t3, $t0, brk17
nbk17:#
#                                  //while (hopPtr < endPtr)
#                                  goto WTest14;
                               j WTest14
begW14:#//                         {
#                                  cout << *hopPtr << ' ' << ' ';
                                   li $v0, 1
                                   lw $a0, 0($t3)
                                   syscall
                                   li $v0, 11
                                   li $a0, ' '
                                   syscall
                                   li $v0, 11
                                   li $a0, ' '
                                   syscall
#                                  ++hopPtr;
                                   addi $t3, $t3, 4
endW14:#//                     }
WTest14:#                      if (hopPtr < endPtr) goto begW14;
                               blt $t3, $t4, begW14
#                              cout << endl;
                               li $v0, 11
                               li $a0, '\n'
                               syscall
#                              //if (endPtr == a1 + used1)
#                              if (endPtr != a1 + used1) goto else18;
                               la $a0, a1
                               sll $t0, $a1, 2
                               add $t0, $t0, $a0
                               bne $t4, $t0, else18
begI18:#//                     {
#                                  cout << comAeStr << 2 << comAfStr;
                                   li $v0, 4
                                   la $a0, comAeStr
                                   syscall
                                   li $v0, 1
                                   li $a0, 2
                                   syscall
                                   li $v0, 4
                                   la $a0, comAfStr
                                   syscall
#                                  hopPtr = a2;
                                   la $t3, a2
#                                  endPtr = hopPtr + used2;
                                   sll $t0, $a2, 2
                                   add $t4, $t0, $t3
#                                  goto endI18;
                                   j endI18
#//                            }
else18:#//                         else
#//                                {
#                                  //if (endPtr == a2 + used2)
#                                  if (endPtr != a2 + used2) goto else19;
                                   la $a0, a2
                                   sll $t0, $a2, 2
                                   add $t0, $t0, $a0
                                   bne $t4, $t0, else19
begI19:#//                         {
#                                          cout << comAeStr << 3 << comAfStr;
                                       li $v0, 4
                                       la $a0, comAeStr
                                       syscall
                                       li $v0, 1
                                       li $a0, 3
                                       syscall
                                       li $v0, 4
                                       la $a0, comAfStr
                                       syscall
#                                          hopPtr = a3;
                                       la $t3, a3
#                                          endPtr = hopPtr + used3;
                                       sll $t0, $a3, 2
                                       add $t4, $t0, $t3
#                                      goto endI19;
                                       j endI19
#//                                }
else19:#//                         else
#//                                {
#                                          //if (endPtr == a3 + used3)
#                                          if (endPtr != a3 + used3) goto endI20;
                                       la $a0, a3
                                       sll $t0, $a3, 2
                                       add $t0, $t0, $a0
                                       bne $t4, $t0, endI20
begI20:#//                                 {
#                                          cout << comAeStr << 4 << comAfStr;
                                           li $v0, 4
                                           la $a0, comAeStr
                                           syscall
                                           li $v0, 1
                                           li $a0, 4
                                           syscall
                                           li $v0, 4
                                           la $a0, comAfStr
                                           syscall
#                                          //if (used4 == 0)
#                                          if (used4 != 0) goto endI21;
                                           bnez $v1, endI21
begI21:#//                                 {
#                                              cout << endl;
                                               li $v0, 11
                                               li $a0, '\n'
                                               syscall
endI21:#//                                 }
#                                          hopPtr = a4;
                                           la $t3, a4
#                                          endPtr = hopPtr + used4;
                                           sll $t0, $v1, 2
                                           add $t4, $t0, $t3
endI20:#//                                 }
endI19:#//                         }
endI18:#//                         }
endW13:#//                 }
WTest13:#                  if (0 == 0) goto begW13;
                           j begW13
```

```
brk17:#
#                          used3 = 0;
                           li $a3, 0
#                          used4 = 0;
                           li $v1, 0
#                          //if ( (minInt & 1) != 0)
#                          if ( (minInt & 1) == 0) goto else22;
                           andi $t0, $t6, 1
                           beqz $t0, else22
begI22:#//                 {
#                              hopPtr = a3;
                               la $t3, a3
#                              used3 = used1 + used2;
                               add $a3, $a1, $a2
#                              goto endI22;
                               j endI22
#//                        }
else22:#//                 else
#//                        {
#                                  hopPtr = a4;
                               la $t3, a4
#                                  used4 = used1 + used2;
                               add $v1, $a1, $a2
endI22:#//                     }
#                          hopPtr1 = a1;
                           la $t1, a1
#                          hopPtr2 = a2;
                           la $t2, a2
#                          endPtr1 = hopPtr1 + used1;
                           sll $t0, $a1, 2
                           add $t9, $t0, $t1
#                          endPtr2 = hopPtr2 + used2;
                           sll $t0, $a2, 2
                           add $t8, $t0, $t2
#                          //while (hopPtr1 < endPtr1 && hopPtr2 < endPtr2)
#                          goto WTest15;
                           j WTest15
begW15:#//                 {
#                                  //if (*hopPtr1 < *hopPtr2)
#                                  if (*hopPtr1 >= *hopPtr2) goto else23;
                               lw $t0, 0($t1)
                               lw $v0, 0($t2)
                               bge $t0, $v0 else23
begI23:#//                         {
#                                  *hopPtr = *hopPtr1;
                                   lw $t0, 0($t1)
                                   sw $t0, 0($t3)
#                                  ++hopPtr1;
                                   addi $t1, $t1, 4
#                                      goto endI23;
                                   j endI23
#//                            }
else23:#//                     else
#//                                {
#                                      *hopPtr = *hopPtr2;
                                   lw $t0, 0($t2)
                                   sw $t0, 0($t3)
#                                  ++hopPtr2;
                                   addi $t2, $t2, 4
endI23:#//                         }
#                              ++hopPtr;
                               addi $t3, $t3, 4
endW15:#//                     }
WTest15:#                  if (hopPtr1 >= endPtr1) goto xitW15;
                           bge $t1, $t9, xitW15
#                          if (hopPtr2 < endPtr2) goto begW15;
                           blt $t2, $t8, begW15
xitW15:#
#                          //while (hopPtr1 < endPtr1)
#                              goto WTest16;
                           j WTest16
begW16:#//                 {
#                                  *hopPtr = *hopPtr1;
                               lw $t0, 0($t1)
                               sw $t0, 0($t3)
#                                  ++hopPtr1;
                               addi $t1, $t1, 4
#                              ++hopPtr;
                               addi $t3, $t3, 4
endW16:#//                 }
WTest16:#                  if (hopPtr1 < endPtr1) goto begW16;
                           blt $t1, $t9, begW16
#
#                              //while (hopPtr2 < endPtr2)
#                              goto WTest17;
                           j WTest17
begW17:#//                     {
#                              *hopPtr = *hopPtr2;
                               lw $t0, 0($t2)
                               sw $t0, 0($t3)
#                              ++hopPtr2;
                               addi $t2, $t2, 4
#                              ++hopPtr;
                               addi $t3, $t3, 4
endW17:#//                 }
WTest17:#                  if (hopPtr2 < endPtr2) goto begW17;
                           blt $t2, $t8, begW17
endI14:#//             }
#                      cout << proA1Str;
                       li $v0, 4
                       la $a0, proA1Str
                       syscall
```

```

#                      hopPtr = a1;
                       la $t3, a1
#                          endPtr = hopPtr + used1;
                       sll $t0, $a1, 2
                       add $t4, $t3, $t0
#
#                      //while (0 == 0)
#                      goto WTest18;
                       j WTest18
begW18:#//             {
#                          //if (hopPtr == a4 + used4 && endPtr == a4 + used4) break;
#                          if (hopPtr != a4 + used4) goto nbk24;
                           la $a0, a4
                           sll $t0, $v1, 2
                           add $t0, $t0, $a0
                           bne $t3, $t0, nbk24
#                          if (endPtr == a4 + used4) goto brk24;
                           la $a0, a4
                           sll $t0, $v1, 2
                           add $t0, $t0, $a0
                           beq $t4, $t0, brk24
nbk24:#
#                          //while (hopPtr < endPtr)
#                          goto WTest19;
                           j WTest19
begW19:#//                 {
#                                  cout << *hopPtr << ' ' << ' ';
                               li $v0, 1
                               lw $a0, 0($t3)
                               syscall
                               li $v0, 11
                               li $a0, ' '
                               syscall
                               li $v0, 11
                               li $a0, ' '
                               syscall
#                                  ++hopPtr;
                               addi $t3, $t3, 4
endW19:#//                 }
WTest19:#                  if (hopPtr < endPtr) goto begW19;
                           blt $t3, $t4, begW19
#                          cout << endl;
                           li $v0, 11
                           li $a0, '\n'
                           syscall
#                          //if (endPtr == a1 + used1)
#                          if (endPtr != a1 + used1) goto else25;
                           la $a0, a1
                           sll $t0, $a1, 2
                           add $t0, $t0, $a0
```

```
                        bne $t4, $t0, else25
begI25:#//              {
#                               cout << comAeStr << 2 << comAfStr;
                            li $v0, 4
                            la $a0, comAeStr
                            syscall
                            li $v0, 1
                            li $a0, 2
                            syscall
                            li $v0, 4
                            la $a0, comAfStr
                            syscall
#                               hopPtr = a2;
                            la $t3, a2
#                           endPtr = hopPtr + used2;
                            sll $t0, $a2, 2
                            add $t4, $t0, $t3
#                           goto endI25;
                            j endI25
#//                     }
else25:#//              else
#//                     {
#                               //if (endPtr == a2 + used2)
#                               if (endPtr != a2 + used2) goto else26;
                            la $a0, a2
                            sll $t0, $a2, 2
                            add $t0, $t0, $a0
                            bne $t4, $t0, else26
begI26:#//                  {
#                               cout << comAeStr << 3 << comAfStr;
                                li $v0, 4
                                la $a0, comAeStr
                                syscall
                                li $v0, 1
                                li $a0, 3
                                syscall
                                li $v0, 4
                                la $a0, comAfStr
                                syscall
#                               hopPtr = a3;
                                la $t3, a3
#                               endPtr = hopPtr + used3;
                                sll $t0, $a3, 2
                                add $t4, $t0, $t3
#                                   goto endI26;
                                j endI26
#//                             }
else26:#//                      else
#//                             {
#                               //if (endPtr == a3 + used3)
#                               if (endPtr != a3 + used3) goto endI27;
```

```
                                  la $a0, a3
                                  sll $t0, $a3, 2
                                  add $t0, $t0, $a0
                                  bne $t4, $t0, endI27
begI27:#//                        {
#                                     cout << comAeStr << 4 << comAfStr;
                                      li $v0, 4
                                      la $a0, comAeStr
                                      syscall
                                      li $v0, 1
                                      li $a0, 4
                                      syscall
                                      li $v0, 4
                                      la $a0, comAfStr
                                      syscall
#                                     //if (used4 == 0)
#                                     if (used4 != 0) goto endI28;
                                      bnez $v1, endI28
begI28:#//                                {
#                                         cout << endl;
                                          li $v0, 11
                                          li $a0, '\n'
                                          syscall
endI28:#//                                }
#                                         hopPtr = a4;
                                      la $t3, a4
#                                         endPtr = hopPtr + used4;
                                      sll $t0, $v1, 2
                                      add $t4, $t0, $t3
endI27:#//                            }
endI26:#//                        }
endI25:#//                {
endW18:#//                }
WTest18:#                 if (0 == 0) goto begW18;
                      j begW18
brk24:#
#
#                     cout << endl;
                      li $v0, 11
                      li $a0, '\n'
                      syscall
#                     cout << dacStr;
                      li $v0, 4
                      la $a0, dacStr
                      syscall
#                     cin >> reply;
                      li $v0, 12
                      syscall
                      move $t6, $v0
#                     cout << endl;
                      li $v0, 11
                      li $a0, '\n'
                      syscall
endDW1:#      }
DWTest1:#   //while (reply != 'n' && reply != 'N');
#             if (reply == 'n') goto xitDW1;
          li $t0, 'n'
          beq $t6, $t0, xitDW1
#             if (reply != 'N') goto begDW1;
          li $t0, 'N'
          bne $t6, $t0, begDW1

xitDW1:
#
#             cout << dlStr;
#         cout << '\n';
          li $v0, 4
          la $a0, dlStr
          syscall
#             cout << byeStr;
#             cout << '\n';
          li $v0, 4
          la $a0, byeStr
          syscall
#         cout << dlStr;
#         cout << '\n';
          li $v0, 4
          la $a0, dlStr
          syscall
#             return 0;
#}
          li $v0, 10      # graceful exit
          syscall
```