```
################################################################################
# Title: Assign02P3                   Author: Ayisha S. R. Sowkathali
# Class: CS 2318-260, Spring 2018     Submitted: 03/26/2018
################################################################################
# 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];
#char reply;
#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* probePtr;

          .data
a1:       .space 48
a2:       .space 48
a3:       .space 48
a4:       .space 48

begA1Str:  .asciiz "\nbeginning a1: "
colSpace:  .asciiz ": """
cpaA1Str:  .asciiz "chkPointA a1: "
proA1Str:  .asciiz "processed a1: "
comA2Str:  .asciiz "\n          a2: "
comA3Str:  .asciiz "\n          a3: "
comA4Str:  .asciiz "\n          a4: "
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 "================================"
byeStr:    .asciiz "bye..."

#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 probePtr
# $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, 4
#         cout << einStr;
          li $v0, 4
          la $a0, einStr
          syscall
#         cout << intNum;
          li $v0, 1
          move $a0, $t5
          syscall
#         cout << ':' << ' ';
          li $v0, 4
          la $a0, colSpace
          syscall
#         cin >> oneInt;
          li $v0, 5
          syscall
          move $t7, $v0

#         //if ( (intNum & 1) != 0 )
#         if ( (intNum & 1) == 0 ) goto else1;
          li $t0, 1                                   # load 1 to $t0
          andi $a0, $t5, 1                            # intNum & 1
          li $t0, 0                                   # load 0 to $t0
          beq $t0, $a0, else1                         # if ( (intNum & 1) == 0 ) goto else1

begI1:
#         *hopPtr1 = oneInt;
          sw  $t7, 0($t1)                             # oneInt in address of hopPtr1
#         ++hopPtr1;
          addi $t1, $t1, 4
#         ++used1;
          addi $a1, $a1, 1
#         goto endI1;
          j endI1

else1:
#         *hopPtr2 = oneInt;
          sw  $t7, 0($t2)                             # oneInt in address of hopPtr2
#         ++hopPtr2;
          addi $t2, $t2, 4
#         ++used2;
          addi $a2, $a2, 1
endI1:
#         //if (intNum == 12)
#         if (intNum != 12) goto else2;
          li $t0, 12
```

```
          bne $t0, $t5, else2
begI2:
#         cout << moStr;
          li $v0, 4
          la $a0, moStr
          syscall
#         cout << 12;
          li $v0, 1
          li $t0, 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:
#         cout << eaiStr;
          li $v0, 4
          la $a0, eaiStr
          syscall
#         cin >> reply;
          li $v0, 12
          syscall
          move $t6, $v0
endI2:
endW1:
WTest1:
#         //if (reply != 'y' && reply != 'Y') goto begW1;
#         if (reply == 'y') goto xitW1;
          li $t0, 'y'
          beq $t6, $t0, xitW1
#         if (reply != 'Y') goto begW1;
          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 = a1 + used1;
          la $t0, a1
          sll $t4, $a1, 2
          add $t4, $t4, $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
          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 << comA2Str;
          li $v0, 4
          la $a0, comA2Str
          syscall
#         hopPtr = a2;
          la $t3, a2
#         //while (hopPtr < endPtr)
#         goto WTest3;
          j WTest3
begW3:
#         cout << *hopPtr << ' ' << ' ';
          li $v0, 1
          lw $a0, 0($t3)
          syscall
          li $v0, 11
          li $a0, ' '
          syscall
          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;
          bgt $a1, $0, begI3
#         if (used2 <= 0) goto else3;
          bge $a2, $0, else3
begI3:
#         hopPtr1 = a1;
          la $t1, a1
#         hopPtr2 = a2;
          la $t2, a2
#         hopPtr3 = a3;
          la $t3, a3
#         hopPtr4 = a4;
          la $t4, a4
#         endPtr1 = a1 + used1;
          sll $t9, $a1, 2
          add $t9, $t9, $t1

#         endPtr2 = a2 + used2;
          sll $t8, $a2, 2
          add $t8, $t8, $t2
#         used3 = 0;
          li $a3, 0
#         used4 = 0;
          li $v1, 0
#         //if (used1 > 0)
#         if (used1 <= 0) goto else4;
          bge $a1, $0, else4
begI4:
#         minInt = *hopPtr1;
          lw $t6, 0($t1)                                # set value of minInt to hopPtr1
#         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 brkI6;
           li $t0, 1                                    # load 1 to $t0
           andi $a0, $t7, 1                             # intNum & 1
           li $t0, 0                                    # load 0 to $t0
           beq $t0, $a0, brkI6                          # if ( (intNum & 1) == 0 ) goto break

#          *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
brkI6:

#          //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 brkI8;
           li $t0, 1                                    # load 1 to $t0
           andi $a0, $t5, 1                             # intNum & 1
           li $t0, 0                                    # load 0 to $t0
           bne $t0, $a0, brkI8                          # if ( (intNum & 1) != 0 ) goto brkI8
#          *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
brkI8:

#          //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;
           sw $t3, 0($t2)
#          *hopPtr4 = *hopPtr1;
           sw $t4, 0($t1)
#          ++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 && hopPtr2 < endPtr2) goto begW4;
#          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;
           li $t0, 1                                    # load 1 to $t0
           andi $a0, $t5, 1                             # intNum & 1
           li $t0, 0                                    # load 0 to $t0
           beq $t0, $a0, else11
begI11:
#          *hopPtr3 = oneInt;
           sw $t7, 0($t3)

#          ++used3;
           addi $a3, $a3, 1
#          ++hopPtr3;
           addi $t3, $t3, 4
#          goto endI11;
           j endI11
else11:
#          *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 elseI13;
           li $t0, 1                                    # load 1 to $t0
```

```
          andi $a0, $t5, 1                              # intNum & 1
          li $t0, 0                                     # load 0 to $t0
          beq $t0, $a0, elseI13
begI13:
#         *hopPtr3 = oneInt;
          sw $t7, 0($t3)
#         ++used3;
          addi $a3, $a3, 1
#         ++hopPtr3;
          addi $t3, $t3, 4
#         goto endI13;
          j endI13
elseI13:
#         *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:
#         used3 = 0;
          li $a3, 0
#         used4 = 0;
          li $v1, 0
endI3:

#         cout << comA3Str;
          li $v0, 4
          la $a0, comA3Str
          syscall
#         hopPtr = a3;
          la $t3, a3
#         endPtr = a3 + used3;
          la $t0, a3
          sll $t4, $a3, 2
          add $t4, $t4, $t0
#         //while (hopPtr < endPtr)
#         goto WTest9;
          j WTest9
begW9:
#         cout << *hopPtr << ' ' << ' ';
          li $v0, 1
          lw $a0, 0($t3)
          syscall
          li $v0, 11
          li $a0, ' '
          syscall
          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 << comA4Str;
          li $v0, 4
          la $a0, comA4Str
          syscall
#         hopPtr = a4;
          la $t3, a4
#         endPtr = a4 + used4;
          sll $t4, $v1, 2
          add $t4, $t4, $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
           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;
           bgt $a1, $0, begI14
#          if (used2 <= 0) goto endI14;
           ble $a2, $0, endI14
begI14:
#          used1 = 0;
           li $a1, 0
#          used2 = 0;
           li $a2, 0
#          hopPtr = a3;
           la $t3, a3
#          endPtr = a3 + used3;
           sll $t4, $a3, 2
           add $t4, $t4, $t3
#          //while (hopPtr < endPtr)
#          goto WTest11;
           j WTest11
begW11:
#          oneInt = *hopPtr;
           lw $t7, 0($t3)
#          //for (probePtr = a1 + used1; probePtr > a1; --probePtr)
#          probePtr = a1 + used1;
           la $t0, a1
           sll $t5, $a1, 2
           add $t5, $t5, $t0
#          goto FTest1;
           j FTest1
begF1:
#          //if ( *(probePtr - 1) <= oneInt ) break;
#          if ( *(probePtr - 1) <= oneInt ) goto brkI15;
           sw $t0, -4($t5)
           ble $t0, $t7, brkI15
#          *probePtr = *(probePtr - 1);
           lw $t0, -4($t5)
           sw $t0, 0($t5)
#          --probePtr;
           addi $t5, $t5, -1
endF1:
FTest1:
#          if (probePtr > a1) goto begF1;
           la $t0, a1
           bgt $t5, $t0, begF1
brkI15:

#          *probePtr = *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 = a4 + used4;
          sll $t4, $v1, 2
          add $t4, $t4, $t3
#         //while (hopPtr < endPtr)
#         goto WTest12;
          j WTest12
begW12:
#         oneInt = *hopPtr;
          lw $t7, 0($t3)

#         //for (probePtr = a2 + used2; probePtr > a2; --probePtr)
#         probePtr = a2 + used2;
          la $t0, a2
          sll $t5, $a2, 2
          add $t5, $t5, $t0

#         goto FTest2;
          j FTest2
begF2:
#         //if ( *(probePtr - 1) <= oneInt ) break;
#         if ( *(probePtr - 1) <= oneInt ) goto brkI16;
#         *probePtr = *(probePtr - 1);
          lw $t0, -4($t5)
          sw $t0, 0($t5)
#         --probePtr;
          addi $t5, $t5, -4
endF2:
FTest2:
#         if (probePtr > a2) goto begF2;
          la $t0, a2
          bgt $t5, $t0, begF2
brkI16:#

#         *probePtr = *hopPtr;
          lw $t0, 0($t3)
          sw $v0, 0($t5)
          move $t0, $v0
#         ++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 = a1 + used1;
          sll $t4, $a1, 2
          add $t4, $t4, $t0

#         //while (hopPtr < endPtr)
#         goto WTest13;
          j WTest13
begW13:
#         cout << *hopPtr << ' ' << ' ';
          li $v0, 1
          lw $a0, 0($t3)
          syscall
          li $v0, 11
          li $a0, ' '
          syscall
          syscall
#         ++hopPtr;
          addi $t3, $t3, 4
endW13:
WTest13:
#         if (hopPtr < endPtr) goto begW13;
```

```
            blt $t3, $t4, begW13

#           cout << endl;
            li $v0, 11
            li $a0, '\n'
            syscall

#           cout << comA2Str;
            li $v0, 4
            la $a0, comA2Str
            syscall
#           hopPtr = a2;
            la $t3, a2
#           endPtr = a2 + used2;
            sll $t4, $a2, 2
            add $t4, $t4, $t0
#           //while (hopPtr < endPtr)
#           goto WTest14;
            j WTest14
begW14:
#           cout << *hopPtr << ' ' << ' ';
            li $v0, 1
            lw $a0, 0($t3)
            syscall
            li $v0, 11
            li $a0, ' '
            syscall
            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
#           cout << comA3Str;
            li $v0, 4
            la $a0, comA3Str
            syscall
#           hopPtr = a3;
            la $t3, a3
#           endPtr = a3 + used3;
            la $t0, a3
            sll $t4, $a2, 2
            add $t4, $t4, $t0
#           //while (hopPtr < endPtr)
#           goto WTest15;
            j WTest15
begW15:
#           cout << *hopPtr << ' ' << ' ';
            li $v0, 1
            lw $a0, 0($t3)
            syscall
            li $v0, 11
            li $a0, ' '
            syscall
            syscall
#           ++hopPtr;
            addi $t3, $t3, 4
endW15:
WTest15:
#           if (hopPtr < endPtr) goto begW15;
            blt $t3, $t4, begW15

#           cout << endl;
            li $v0, 11
            li $a0, '\n'
            syscall
#           cout << comA4Str;
            li $v0, 4
            la $a0, comA4Str
            syscall
#           hopPtr = a4;
```

```
          la $t3, a4
#         endPtr = a4 + used4;
          sll $t4, $v1, 2
          add $t4, $t4, $t3
#         //while (hopPtr < endPtr)
#         goto WTest16;
          j WTest16
begW16:
#         cout << *hopPtr << ' ' << ' ';
          li $v0, 1
          lw $a0, 0($t3)
          syscall
          li $v0, 11
          li $a0, ' '
          syscall
          syscall
#         ++hopPtr;
          addi $t3, $t3, 4
endW16:
WTest16:
#         if (hopPtr < endPtr) goto begW16;
          blt $t3, $t4, begW16

#         cout << endl;
          li $v0, 11
          li $a0, '\n'
          syscall
#         used3 = 0;
          li $a3, 0
#         used4 = 0;
          li $v1, 0
#         //if ( (minInt & 1) != 0)
#         if ( (minInt & 1) == 0) goto else17;
          li $t0, 1                                     # load 1 to $t0
          andi $a0, $t6, 1                              # minInt & 1
          li $t0, 0                                     # load 0 to $t0
          beq $t0, $a0, else17
begI17:
#         hopPtr = a3;
          la $t3, a3
#         used3 = used1 + used2;
          add $a3, $a1, $a2
#         goto endI17;
          j endI17
else17:
#         hopPtr = a4;
          la $t3, a4
#         used4 = used1 + used2;
          add $v1, $a1, $a2
endI17:
#         hopPtr1 = a1;
          la $t1, a1
#         hopPtr2 = a2;
          la $t2, a2
#         endPtr1 = a1 + used1;
          sll $t9, $a1, 2
          add $t9, $t9, $t1
#         endPtr2 = a2 + used2;
          sll $t8, $a2, 2
          add $t8, $t8, $t2

#         //while (hopPtr1 < endPtr1 && hopPtr2 < endPtr2)
#         goto WTest17;
          j WTest7
begW17:
#         //if (*hopPtr1 < *hopPtr2)
#         if (*hopPtr1 >= *hopPtr2) goto elseI18;
          lw $a0, 0($t1)
          lw $t0, 0($t2)
          bge $a0, $t0, elseI18
begI18:
#         *hopPtr = *hopPtr1;
          lw $a0, 0($t3)
          sw $t0, 0($t1)
          move $a0, $t0
#         ++hopPtr1;
          addi $t1, $t1, 4
```

```
#          goto endI18;
           j endI18
elseI18:
#          *hopPtr = *hopPtr2;
           lw $a0, 0($t3)
           sw $t0, 0($t2)
           move $a0, $t0
#          ++hopPtr2;
           addi $t2, $t2, 4
endI18:
#          ++hopPtr;
           addi $t3, $t3, 4
endW17:
WTest17:
#          //if (hopPtr1 < endPtr1 && hopPtr2 < endPtr2) goto begW17;
#          if (hopPtr1 >= endPtr1) goto xitW17;
           bge $t1, $t9, xitW17
#          if (hopPtr2 < endPtr2) goto begW17;
           blt $t2, $t8, begW17
xitW17:

#          //while (hopPtr1 < endPtr1)
#          goto WTest18;
           j WTest18
begW18:
#          *hopPtr = *hopPtr1;
           lw $a0, 0($t3)
           sw $t0, 0($t1)
           move $a0, $t0
#          ++hopPtr1;
           addi $t1, $t1, 4
#          ++hopPtr;
           addi $t3, $t3, 4
endW18:
WTest18:
#          if (hopPtr1 < endPtr1) goto begW18;
           ble $t1, $t9, begW18

#          //while (hopPtr2 < endPtr2)
#          goto WTest19;
           j WTest19
begW19:
#          *hopPtr = *hopPtr2;
           lw $a0, 0($t3)
           sw $t0, 0($t2)
           move $a0, $t0
#          ++hopPtr2;
           addi $t2, $t2, 4
#          ++hopPtr;
           addi $t3, $t3, 4
endW19:
WTest19:
#          if (hopPtr2 < endPtr2) goto begW19;
           ble $t2, $t8, begW19

endI14:
#          cout << proA1Str;
           li $v0, 4
           la $a0, proA1Str
           syscall
#          hopPtr = a1;
           la $t3, a1
#          endPtr = a1 + used1;
           sll $t4, $a1, 2
           add $t4, $t4, $t3
#          //while (hopPtr < endPtr)
#          goto WTest20;
           j WTest20
begW20:
#          cout << *hopPtr << ' ' << ' ';
           li $v0, 1
           lw $a0, 0($t3)
           syscall
           li $v0, 11
           li $a0, ' '
           syscall
           syscall
```

```
#         ++hopPtr;
          addi $t3, $t3, 4
endW20:
WTest20:
#         if (hopPtr < endPtr) goto begW20;
          ble $t3, $t4, begW20

#         cout << endl;
          li $v0, 11
          li $a0, '\n'
          syscall
#         cout << comA2Str;
          li $v0, 4
          la $a0, comA2Str
          syscall
#         hopPtr = a2;
          la $t3, a2
#         endPtr = a2 + used2;
          sll $t4, $a2, 2
          add $t4, $t4, $t3

#         //while (hopPtr < endPtr)
#         goto WTest21;
          j WTest21
begW21:
#         cout << *hopPtr << ' ' << ' ';
          li $v0, 1
          lw $a0, 0($t3)
          syscall
          li $v0, 11
          li $a0, ' '
          syscall
          syscall
#         ++hopPtr;
          addi $t3, $t3, 4
endW21:
WTest21:
#         if (hopPtr < endPtr) goto begW21;
          ble $t3, $t4, begW21

#         cout << endl;
          li $v0, 11
          li $a0, '\n'
          syscall
#         cout << comA3Str;
          li $v0, 4
          la $a0, comA3Str
          syscall
#         hopPtr = a3;
          la $t3, a3
#         endPtr = a3 + used3;
          sll $t4, $a3, 2
          add $t4, $t4, $t3
#         //while (hopPtr < endPtr)
#         goto WTest22;
          j WTest22
begW22:
#         cout << *hopPtr << ' ' << ' ';
          li $v0, 1
          lw $a0, 0($t3)
          syscall
          li $v0, 11
          li $a0, ' '
          syscall
          syscall
#         ++hopPtr;
          addi $t3, $t3, 4
endW22:
WTest22:
#         if (hopPtr < endPtr) goto begW22;
          ble $t3, $t4, begW22

#         cout << endl;
          li $v0, 11
          li $a0, '\n'
          syscall
#         cout << comA4Str;
```

```
            li $v0, 4
            la $a0, comA4Str
            syscall
#           hopPtr = a4;
            la $t3, a4
#           endPtr = a4 + used4;
            la $t0, a4
            sll $t4, $t0, 2
            add $t4, $t4, $t0

#           //while (hopPtr < endPtr)
#           goto WTest23;
            j WTest23
begW23:
#           cout << *hopPtr << ' ' << ' ';
            li $v0, 1
            lw $a0, 0($t3)
            syscall
            li $v0, 11
            li $a0, ' '
            syscall
            syscall
#           ++hopPtr;
            addi $t3, $t3, 4
endW23:
WTest23:
#           if (hopPtr < endPtr) goto begW23;
            blt $t3, $t4, begW23

#           cout << endl;
            li $v0, 11
            li $a0, '\n'
            syscall
#           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:
#           //while (reply != 'n' && reply != 'N');
DWTest1:
#           //if (reply != 'n' && reply != 'N') goto begDW1;
#           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;
            li $v0, 4
            la $a0, dlStr
            syscall
#           cout << '\n';
            li $v0, 11
            li $a0, '\n'
            syscall
#           cout << byeStr;
            li $v0, 4
            la $a0, byeStr
            syscall
#           cout << '\n';
            li $v0, 11
            li $a0, '\n'
            syscall
#           cout << dlStr;
```

```
            li $v0, 4
            la $a0, dlStr
            syscall
#           cout << '\n';
            li $v0, 11
            li $a0, '\n'
            syscall

#           return 0;
            li $v0, 10
            syscall
#}
```