```
1:  ###########################################################################
2:  # Title: Assign02P3                    Author: Ethan West
3:  # Class: CS 2318-25?, Spring 2024     Submitted: 04/18/2024
4:  ###########################################################################
5:  # Program: MIPS tranlation of a given C++ program
6:  ###########################################################################
7:  # Pseudocode description: supplied a2p2_SampSoln.cpp
8:  ###########################################################################
9:
10: ##include <iostream>
11: #using namespace std;
12:
13: #int a1[12], a2[12], a3[12], a4[12];
14: #int used1, used2, used3, used4, minInt, intNum, oneInt;
15: #int* hopPtr;
16: #int* hopPtr1;
17: #int* hopPtr2;
18: #int* hopPtr3;
19: #int* hopPtr4;
20: #int* endPtr;
21: #int* endPtr1;
22: #int* endPtr2;
23: #int* iPtr;
24: #char reply;
25: #char begA1Str[] = "beginning a1: ";
26: #char cpaA1Str[] = "chkPointA a1: ";
27: #char proA1Str[] = "processed a1: ";
28: #char comAeStr[] = "          a";
29: #char comAfStr[] = ": ";
30: #char einStr[]   = "Enter integer #";
31: #char moStr[]    = "Max of ";
32: #char ieStr[]    = " ints entered...";
33: #char eaiStr[]   = "End adding ints? (y or Y = yes, others = no) ";
34: #char dacStr[]   = "Do another case? (n or N = no, others = yes) ";
35: #char dlStr[]    = "================================";
36: #char byeStr[]   = "bye...";
37:
38:             .data
39: a1:         .space 40
40: a2:         .space 40
41: a3:         .space 40
42: a4:         .space 40
43: begA1Str:   .asciiz "\nbeginning a1: "
44: cpaA1Str:   .asciiz "chkPointA a1: "
45: proA1Str:   .asciiz "processed a1: "
46: comAeStr:   .asciiz "     a"
47: comAfStr:   .asciiz ": "
48: einStr:     .asciiz "\nEnter integer #"
49: moStr:      .asciiz "Max of "
50: ieStr:      .asciiz " ints entered..."
51: eaiStr:     .asciiz "End adding ints? (y or Y = yes, others = no) "
```

```
52: dacStr:     .asciiz "Do another case? (n or N = no, others = yes) "
53: dlStr:      .asciiz "\n================================"
54: byeStr:     .asciiz "\nbye..."
55:
56: #int main()
57: #{
58:             .text
59:             .globl main
60: main:
61:
62: ################################################
63: # Register usage:
64: #################
65: # $a0: short-lived holder 3
66: # $a1: used1
67: # $a2: used2
68: # $a3: used3
69: # $v1: used4
70: # $t0: short-lived holder 1
71: # $t1: hopPtr1
72: # $t2: hopPtr2
73: # $t3: hopPtr3 or hopPtr
74: # $t4: hopPtr4 or endPtr
75: # $t5: intNum or iPtr
76: # $t6: minInt or reply
77: # $t7: oneInt
78: # $t8: endPtr2
79: # $t9: endPtr1
80: # $v0: short-lived holder 2
81: ################################################
82:
83: #           //do
84: begDW1:#    {
85: #               intNum = 0;
86:                 li $t5, 0
87: #               used1 = 0;
88:                 li $a1, 0
89: #               used2 = 0;
90:                 li $a2, 0
91: #               hopPtr1 = a1;
92:                 la $t1, a1
93: #               hopPtr2 = a2;
94:                 la $t2, a2
95: #               cout << eaiStr;
96:                 li $v0, 4
97:                 la $a0, eaiStr
98:                 syscall
99: #               cin >> reply;
100:                li $v0, 12
101:                syscall
102:                move $t6, $v0
```

```
103:
104: #                  //while (reply != 'y' && reply != 'Y')
105: #                goto WTest1;
106:                  j WTest1
107: begW1:#             {
108: #                  ++intNum;
109:                    addi $t5, $t5, 1
110: #                   cout << einStr;
111:                    li $v0, 4
112:                    la $a0, einStr
113:                    syscall
114: #                   cout << intNum;
115:                    li $v0, 1
116:                    move $a0, $t5
117:                    syscall
118: #                   cout << ':' << ' ';
119:                    li $v0, 4
120:                    la $a0, comAfStr
121:                    syscall
122: #                   cin >> oneInt;
123:                    li $v0, 5
124:                    syscall
125:                    move $t7, $v0
126: #                       //if ( (intNum & 1) != 0 )
127: #                    if ( (intNum & 1) == 0 ) goto else1;
128:                    andi $t0, $t5, 1
129:                    beqz $t0, else1
130: begI1:#             {
131: #                       *hopPtr1 = oneInt;
132:                        sw $t7, 0($t1)
133: #                      ++hopPtr1;
134:                        addi $t1, $t1, 4
135: #                      ++used1;
136:                        addi $a1, $a1, 1
137: #                      goto endI1;
138:                        j endI1
139: #//                    }
140: else1:#//               else
141: #//                    {
142: #                         *hopPtr2 = oneInt;
143:                          sw $t7, 0($t2)
144: #                        ++hopPtr2;
145:                          addi $t2, $t2, 4
146: #                        ++used2;
147:                          addi $a2, $a2, 1
148: endI1:#//                 }
149: #                      //if (intNum == 12)
150: #                      if (intNum != 12) goto else2;
151:                        li $t0, 12
152:                        bne $t5, $t0, else2
153: begI2:#//               {
```

```
154: #                          cout << moStr;
155:                            li $v0, 4
156:                            la $a0, moStr
157:                            syscall
158: #                          cout << 12;
159:                            li $v0, 1
160:                            li $a0, 12
161:                            syscall
162: #                          cout << ieStr;
163:                            li $v0, 4
164:                            la $a0, ieStr
165:                            syscall
166: #                          cout << endl;
167:                            li $v0, 11
168:                            li $a0, '\n'
169:                            syscall
170: #                          reply = 'y';
171:                            li $t6, 'y'
172: #                              goto endI2;
173:                            j endI2
174: #//                      }
175: else2:#//               else
176: #//                      {
177: #                          cout << eaiStr;
178:                            li $v0, 4
179:                            la $a0, eaiStr
180:                            syscall
181: #                          cin >> reply;
182:                            li $v0, 12
183:                            syscall
184:                            move $t6, $v0
185: endI2:#//               }
186: endW1:#//               }
187: WTest1:#
188: #                          if (reply == 'y') goto xitW1;
189: #                          if (reply != 'Y') goto begW1;
190:                         li $t0, 'y'
191:                         beq $t6, $t0, xitW1
192:                         li $t0, 'Y'
193:                         bne $t6, $t0, begW1
194: xitW1:#
195: #                        cout << endl;
196:                         li $v0, 11
197:                         li $a0, '\n'
198:                         syscall
199: #                        cout << begA1Str;
200:                         li $v0, 4
201:                         la $a0, begA1Str
202:                         syscall
203: #                        hopPtr = a1;
204:                         la $t3, a1
```

```
205: #                      endPtr = hopPtr + used1;
206:                        sll $t0, $a1, 2
207:                        add $t4, $t3, $t0
208: #                      //while (hopPtr < endPtr)
209: #                      goto WTest2;
210:                        j WTest2
211: begW2:#//                  {
212: #                          cout << *hopPtr << ' ' << ' ';
213:                            li $v0, 1
214:                            lw $a0, 0($t3)
215:                            syscall
216:                            li $v0, 11
217:                            li $a0, ' '
218:                            syscall
219:                            li $v0, 11
220:                            li $a0, ' '
221:                            syscall
222: #                          ++hopPtr;
223:                            addi $t3, $t3, 4
224: endW2:#//                  }
225: WTest2:#                   if (hopPtr < endPtr) goto begW2
226:                        blt $t3, $t4, begW2
227: #
228: #                      cout << endl;
229:                        li $v0, 11
230:                        li $a0, '\n'
231:                        syscall
232: #
233: #                      cout << comAeStr << 2 << comAfStr;
234:                        li $v0, 4
235:                        la $a0, comAeStr
236:                        syscall
237:                        li $v0, 1
238:                        li $a0, 2
239:                        syscall
240:                        li $v0, 4
241:                        la $a0, comAfStr
242:                        syscall
243: #                      hopPtr = a2;
244:                        la $t3, a2
245: #                      endPtr = hopPtr + used2;
246:                        sll $t0, $a2, 2
247:                        add $t4, $t3, $t0
248: #                      //while (hopPtr < endPtr)
249: #                      goto WTest3;
250:                        j WTest3
251: begW3:#//                  {
252: #                          cout << *hopPtr << ' ' << ' ';
253:                            li $v0, 1
254:                            lw $a0, 0($t3)
255:                            syscall
```

```
256:                            li $v0, 11
257:                            li $a0, ' '
258:                            syscall
259:                            li $v0, 11
260:                            li $a0, ' '
261:                            syscall
262: #                          ++hopPtr;
263:                            addi $t3, $t3, 4
264: endW3:#//                  }
265: WTest3:#                   if (hopPtr < endPtr) goto begW3;
266:                          blt $t3, $t4, begW3
267: #                          cout << endl;
268:                          li $v0, 11
269:                          li $a0, '\n'
270:                          syscall
271: #                        //if (used1 > 0 || used2 > 0)
272: #                        if (used1 > 0) goto begI3;
273:                          bgtz $a1, begI3
274: #                        if (used2 <= 0) goto else3;
275:                          blez $a2, else3
276: begI3:#//                {
277: #                            hopPtr1 = a1;
278:                              la $t1, a1
279: #                            hopPtr2 = a2;
280:                              la $t2, a2
281: #                                hopPtr3 = a3;
282:                              la $t3, a3
283: #                                hopPtr4 = a4;
284:                              la $t4, a4
285: #                            endPtr1 = hopPtr1 + used1;
286:                              sll $t0, $a1, 2
287:                              add $t9, $t1, $t0
288: #                            endPtr2 = hopPtr2 + used2;
289:                              sll $t0, $a2, 2
290:                              add $t8, $t2, $t0
291: #                            used3 = 0;
292:                              li $a3, 0
293: #                            used4 = 0;
294:                              li $v1, 0
295: #                            //if (used1 > 0)
296: #                            if (used1 <= 0) goto else4;
297:                              blez $a1, else4
298: begI4:#//                     {
299: #                                minInt = *hopPtr1;
300:                                  lw $t6, 0($t1)
301: #                                goto endI4;
302:                                  j endI4
303: #//                           }
304: else4:#//                        else
305: #//                            {
306: #                                minInt = *hopPtr2;
```

```
307:                                lw $t6, 0($t2)
308: endI4:#//                      }
309: #                              //while (hopPtr1 < endPtr1 && hopPtr2 < endPtr2)
310: #                              goto WTest4;
311:                               j WTest4
312: begW4:#//                      {
313: #                              //while (hopPtr1 < endPtr1)
314: #                                  goto WTest5;
315:                                j WTest5
316: begW5:#//                      {
317: #                               oneInt = *hopPtr1;
318:                                lw $t7, 0($t1)
319: #                               //if (oneInt < minInt)
320: #                              if (oneInt >= minInt) goto endI5;
321:                                bge $t7, $t6, endI5
322: begI5:#//                       {
323: #                                      minInt = oneInt;
324:                                   move $t6, $t7
325: endI5:#//                       }
326: #                               //if ( (oneInt & 1) == 0 ) break;
327: #                               if ( (oneInt & 1) == 0 ) goto brk6;
328:                                andi $t0, $t7, 1
329:                                beqz $t0, brk6
330: #                               *hopPtr3 = oneInt;
331:                                sw $t7, 0($t3)
332: #                               ++used3;
333:                                addi $a3, $a3, 1
334: #                               ++hopPtr1;
335:                                addi $t1, $t1, 4
336: #                               ++hopPtr3;
337:                                addi $t3, $t3, 4
338: endW5:#//                       }
339: WTest5:#                       if (hopPtr1 < endPtr1) goto begW5;
340:                                blt $t1, $t9, begW5
341: brk6:#
342: #                              //while (hopPtr2 < endPtr2)
343: #                              goto WTest6;
344:                                j WTest6
345: begW6:#//                      {
346: #                                  oneInt = *hopPtr2;
347:                                    lw $t7, 0($t2)
348: #                                  //if (oneInt < minInt)
349: #                                  if (oneInt >= minInt) goto endI7;
350:                                    bge $t7, $t6, endI7
351: begI7:#//                           {
352: #                                      minInt = oneInt;
353:                                        move $t6, $t7
354: endI7:#//                           }
355: #                                  //if ( (oneInt & 1) != 0 ) break;
356: #                                  if ( (oneInt & 1) != 0 ) goto brk8;
357:                                    andi $t0, $t7, 1
```

```
358:                                    bnez $t0, brk8
359: #                                  *hopPtr4 = oneInt;
360:                                    sw $t7, 0($t4)
361: #                                  ++used4;
362:                                    addi $v1, $v1,  1
363: #                                  ++hopPtr2;
364:                                    addi $t2, $t2, 4
365: #                                  ++hopPtr4;
366:                                    addi $t4, $t4, 4
367: endW6:#//                       }
368: WTest6:#                        if (hopPtr2 < endPtr2) goto begW6;
369:                                 blt $t2, $t8, begW6
370: brk8:#
371: #                                //if (hopPtr1 < endPtr1 && hopPtr2 < endPtr2)
372: #                                if (hopPtr1 >= endPtr1) goto endI9;
373:                                 bge $t1, $t9, endI9
374: #                                if (hopPtr2 >= endPtr2) goto endI9;
375:                                 bge $t2, $t8, endI9
376: begI9:#//                       {
377: #                                  *hopPtr3 = *hopPtr2;
378:                                    lw $t0, 0($t2)
379:                                    sw $t0, 0($t3)
380: #                                  *hopPtr4 = *hopPtr1;
381:                                    lw $t0, 0($t1)
382:                                    sw $t0, 0($t4)
383: #                                  ++used3;
384:                                    addi $a3, $a3, 1
385: #                                  ++used4;
386:                                    addi $v1, $v1, 1
387: #                                  ++hopPtr1;
388:                                    addi $t1, $t1, 4
389: #                                  ++hopPtr2;
390:                                    addi $t2, $t2, 4
391: #                                  ++hopPtr3;
392:                                    addi $t3, $t3, 4
393: #                                  ++hopPtr4;
394:                                    addi $t4, $t4, 4
395: endI9:#//                       }
396: endW4:#//                   }
397: WTest4:#                    if (hopPtr1 >= endPtr1) goto xitW4;
398:                             bge $t1, $t9, xitW4
399: #                           if (hopPtr2 < endPtr2) goto begW4;
400:                             blt $t2, $t8, begW4
401: xitW4:#
402: #
403: #                               //while (hopPtr1 < endPtr1)
404: #                           goto WTest7;
405:                             j WTest7
406: begW7:#//                       {
407: #                              oneInt = *hopPtr1;
408:                                lw $t7, 0($t1)
```

```
409: #                                  //if (oneInt < minInt)
410: #                                  if (oneInt >= minInt) goto endI10;
411:                                    bge $t7, $t6, endI10
412: begI10:#//                             {
413: #                                          minInt = oneInt;
414:                                        move $t6, $t7
415: endI10:#//                         }
416: #                                  //if ( (oneInt & 1) != 0 )
417: #                                  if ( (oneInt & 1) == 0 ) goto else11;
418:                                    andi $t0, $t7, 1
419:                                    beqz $t0, else11
420: begI11:#//                             {
421: #                                      *hopPtr3 = oneInt;
422:                                        sw $t7, 0($t3)
423: #                                      ++used3;
424:                                        addi $a3, $a3, 1
425: #                                      ++hopPtr3;
426:                                        addi $t3, $t3, 4
427: #                                          goto endI11;
428:                                        j endI11
429: #//                                 }
430: else11:#//                              else
431: #//                                 {
432: #                                      *hopPtr4 = oneInt;
433:                                        sw $t7, 0($t4)
434: #                                      ++used4;
435:                                        addi $v1, $v1, 1
436: #                                      ++hopPtr4;
437:                                        addi $t4, $t4, 4
438: endI11:#//                          }
439: #                                          ++hopPtr1;
440:                                        addi $t1, $t1, 4
441: endW7:#//                       }
442: WTest7:#                       if (hopPtr1 < endPtr1) goto begW7;
443:                                blt $t1, $t9, begW7
444: #
445: #                              //while (hopPtr2 < endPtr2)
446: #                                  goto WTest8;
447:                                j WTest8
448: begW8:#//                       {
449: #                                      oneInt = *hopPtr2;
450:                                    lw $t7, 0($t2)
451: #                                      //if (oneInt < minInt)
452: #                                      if (oneInt >= minInt) goto endI12;
453:                                    bge $t7, $t6, endI12
454: begI12:#//                             {
455: #                                      minInt = oneInt;
456:                                        move $t6, $t7
457: endI12:#//                             }
458: #                                      //if ( (oneInt & 1) != 0 )
459: #                                      if ( (oneInt & 1) == 0 ) goto else13;
```

```
460:                                andi $t0, $t7, 1
461:                                beqz $t0, else13
462: begI13:#//                         {
463: #                                  *hopPtr3 = oneInt;
464:                                    sw $t7, 0($t3)
465: #                                  ++used3;
466:                                    addi $a3, $a3, 1
467: #                                  ++hopPtr3;
468:                                    addi $t3, $t3, 4
469: #                                      goto endI13;
470:                                    j endI13
471: #//                                 }
472: else13:#//                         else
473: #//                                 {
474: #                                  *hopPtr4 = oneInt;
475:                                    sw $t7, 0($t4)
476: #                                  ++used4;
477:                                    addi $v1, $v1, 1
478: #                                  ++hopPtr4;
479:                                    addi $t4, $t4, 4
480: endI13:#//                      }
481: #                                  ++hopPtr2;
482:                                 addi $t2, $t2, 4
483: endW8:#//                   }
484: WTest8:#                       if (hopPtr2 < endPtr2) goto begW8;
485:                            blt $t2, $t8, begW8
486: #                          goto endI3;
487:                            j endI3
488:
489: #//                    }
490: else3:#//              else
491: #//                    {
492: #                       used3 = 0;
493:                        li $a3, 0
494: #                      used4 = 0;
495:                        li $v1, 0
496: endI3:#//              }
497: #                      cout << comAeStr << 3 << comAfStr;
498:                        li $v0, 4
499:                        la $a0, comAeStr
500:                        syscall
501:                        li $v0, 1
502:                        li $a0, 3
503:                        syscall
504:                        li $v0, 4
505:                        la $a0, comAfStr
506:                        syscall
507:
508: #                      hopPtr = a3;
509:                        la $t3, a3
510: #                      endPtr = hopPtr + used3;
```

```
511:                        sll $t0, $a3, 2
512:                        add $t4, $t0, $t3
513: #                      //while (hopPtr < endPtr)
514: #                      goto WTest9;
515:                        j WTest9
516: begW9:#//              {
517: #                              cout << *hopPtr << ' ' << ' ';
518:                            li $v0, 1
519:                            lw $a0, 0($t3)
520:                            syscall
521:                            li $v0, 11
522:                            li $a0, ' '
523:                            syscall
524:                            li $v0, 11
525:                            li $a0, ' '
526:                            syscall
527:
528: #                              ++hopPtr;
529:                            addi $t3, $t3, 4
530: endW9:#//                   }
531: WTest9:#                    if (hopPtr < endPtr) goto begW9;
532:                        blt $t3, $t4, begW9
533: #                      cout << endl;
534:                        li $v0, 11
535:                        li $a0, '\n'
536:                        syscall
537: #                      cout << comAeStr << 4 << comAfStr;
538:                        li $v0, 4
539:                        la $a0, comAeStr
540:                        syscall
541:                        li $v0, 1
542:                        li $a0, 4
543:                        syscall
544:                        li $v0, 4
545:                        la $a0, comAfStr
546:                        syscall
547:
548: #                      hopPtr = a4;
549:                        la $t3, a4
550: #                      endPtr = hopPtr + used4;
551:                        sll $t0, $v1, 2
552:                        add $t4, $t0, $t3
553: #                      //while (hopPtr < endPtr)
554: #                      goto WTest10;
555:                        j WTest10
556: begW10:#//              {
557: #                          cout << *hopPtr << ' ' << ' ';
558:                            li $v0, 1
559:                            lw $a0, 0($t3)
560:                            syscall
561:                            li $v0, 11
```

```
562:                            li $a0, ' '
563:                            syscall
564:                            li $v0, 11
565:                            li $a0, ' '
566:                            syscall
567: #                          ++hopPtr;
568:                            addi $t3, $t3, 4
569: endW10:#//                 }
570: WTest10:#                  if (hopPtr < endPtr) goto begW10;
571:                            blt $t3, $t4, begW10
572: #                          cout << endl;
573:                            li $v0, 11
574:                            li $a0, '\n'
575:                            syscall
576: #                          //if (used1 > 0 || used2 > 0)
577: #                          if (used1 > 0) goto begI14;
578:                            bgtz $a1, begI14
579: #                          if (used2 <= 0) goto endI14;
580:                            blez $a2, endI14
581: begI14:#//                  {
582: #                                  used1 = 0;
583:                                li $a1, 0
584: #                                  used2 = 0;
585:                                li $a2, 0
586: #                                  hopPtr = a3;
587:                                la $t3, a3
588: #                                  endPtr = hopPtr + used3;
589:                                sll $t0, $a3, 2
590:                                add $t4, $t0, $t3
591: #                                  //while (hopPtr < endPtr)
592: #                                goto WTest11;
593:                                j WTest11
594: begW11:#//                      {
595: #                                  oneInt = *hopPtr;
596:                                    lw $t7, 0($t3)
597: #                                  //for (iPtr = a1 + used1; iPtr > a1; --iPtr)
598: #                                      iPtr = a1 + used1;
599:                                    la $a0, a1
600:                                    sll $t0, $a1, 2
601:                                    add $t5, $t0, $a0
602: #                                      goto FTest1;
603:                                    j FTest1
604: begF1:#//                           {
605: #                                      //if ( *(iPtr - 1) <= oneInt ) break;
606: #                                      if ( *(iPtr - 1) <= oneInt ) goto brk15;
607:                                        lw $t0, -4($t5)
608:                                        ble $t0, $t7, brk15
609: #                                      *iPtr = *(iPtr - 1);
610:                                        lw $t0, -4($t5)
611:                                        sw $t0, 0($t5)
612: #                                      --iPtr;
```

```
613:                                        addi $t5, $t5, -4
614: endF1:#//                          }
615: FTest1:#                           if (iPtr > a1) goto begF1;
616:                                    la $t0, a1
617:                                    bgt $t5, $t0, begF1
618: brk15:#
619: #                                  *iPtr = *hopPtr;
620:                                    lw $t0, 0($t3)
621:                                    sw $t0, 0($t5)
622: #                                  ++used1;
623:                                    addi $a1, $a1, 1
624: #                                  ++hopPtr;
625:                                    addi $t3, $t3, 4
626: endW11:#//                          }
627: WTest11:#                       if (hopPtr < endPtr) goto begW11;
628:                                 blt $t3, $t4, begW11
629: #
630: #                                    hopPtr = a4;
631:                                 la $t3, a4
632: #                               endPtr = hopPtr + used4;
633:                                 sll $t0, $v1, 2
634:                                 add $t4, $t0, $t3
635: #                               //while (hopPtr < endPtr)
636: #                               goto WTest12;
637:                                 j WTest12
638: begW12:#//                      {
639: #                                    oneInt = *hopPtr;
640:                                      lw $t7, 0($t3)
641: #                                    //for (iPtr = a2 + used2; iPtr > a2; --iPtr)
642: #                                        iPtr = a2 + used2;
643:                                      la $t5, a2
644:                                      sll $t0, $a2, 2
645:                                      add $t5, $t5, $t0
646: #                                    goto FTest2;
647:                                      j FTest2
648: begF2:#//                            {
649: #                                        //if ( *(iPtr - 1) <= oneInt ) break;
650: #                                        if ( *(iPtr - 1) <= oneInt ) goto brk16;
651:                                          lw $t0, -4($t5)
652:                                          ble $t0, $t7, brk16
653: #                                        *iPtr = *(iPtr - 1);
654:                                          lw $t0, -4($t5)
655:                                          sw $t0, 0($t5)
656: #                                        --iPtr;
657:                                          addi $t5, $t5, -4
658: endF2:#//                                }
659: FTest2:#                                if (iPtr > a2) goto begF2;
660:                                      la $t0, a2
661:                                      bgt $t5, $t0, begF2
662: brk16:#
663: #
```

```
664: #                              *iPtr = *hopPtr;
665:                                lw $t0, 0($t3)
666:                                sw $t0, 0($t5)
667: #                              ++used2;
668:                                addi $a2, $a2, 1
669: #                              ++hopPtr;
670:                                addi $t3, $t3, 4
671: endW12:#//                     }
672: WTest12:#                    if (hopPtr < endPtr) goto begW12;
673:                               blt $t3, $t4, begW12
674: #                              cout << cpaA1Str;
675:                               li $v0, 4
676:                               la $a0, cpaA1Str
677:                               syscall
678: #                              hopPtr = a1;
679:                               la $t3, a1
680: #                              endPtr = hopPtr + used1;
681:                               sll $t0, $a1, 2
682:                               add $t4, $t0, $t3
683: #
684: #                              //while (0 == 0)
685: #                              goto WTest13;
686:                               j WTest13
687: begW13:#//                         {
688: #                                  //if (hopPtr == a4 + used4 && endPtr == a4 + used4
) break;
689: #                                  ////if (hopPtr == a4 + used4 && endPtr == a4 + use
d4) goto brk17;
690: #                                  if (hopPtr != a4 + used4) goto nbk17;
691:                                    la $a0, a4
692:                                    sll $t0, $v1, 2
693:                                    add $t0, $t0, $a0
694:                                    bne $t3, $t0, nbk17
695: #                                  if (endPtr == a4 + used4) goto brk17;
696:                                    la $a0, a4
697:                                    sll $t0, $v1, 2
698:                                    add $t0, $t0, $a0
699:                                    beq $t3, $t0, brk17
700: nbk17:#
701: #                                  //while (hopPtr < endPtr)
702: #                                  goto WTest14;
703:                                    j WTest14
704: begW14:#//                             {
705: #                                  cout << *hopPtr << ' ' << ' ';
706:                                    li $v0, 1
707:                                    lw $a0, 0($t3)
708:                                    syscall
709:                                    li $v0, 11
710:                                    li $a0, ' '
711:                                    syscall
712:                                    li $v0, 11
```

```
713:                                    li $a0, ' '
714:                                    syscall
715: #                                  ++hopPtr;
716:                                    addi $t3, $t3, 4
717: endW14:#//                        }
718: WTest14:#                         if (hopPtr < endPtr) goto begW14;
719:                                   blt $t3, $t4, begW14
720: #                                  cout << endl;
721:                                   li $v0, 11
722:                                   li $a0, '\n'
723:                                   syscall
724: #                                 //if (endPtr == a1 + used1)
725: #                                 if (endPtr != a1 + used1) goto else18;
726:                                   la $a0, a1
727:                                   sll $t0, $a1, 2
728:                                   add $t0, $t0, $a0
729:                                   bne $t4, $t0, else18
730: begI18:#//                        {
731: #                                     cout << comAeStr << 2 << comAfStr;
732:                                       li $v0, 4
733:                                       la $a0, comAeStr
734:                                       syscall
735:                                       li $v0, 1
736:                                       li $a0, 2
737:                                       syscall
738:                                       li $v0, 4
739:                                       la $a0, comAfStr
740:                                       syscall
741: #                                     hopPtr = a2;
742:                                       la $t3, a2
743: #                                     endPtr = hopPtr + used2;
744:                                       sll $t0, $a2, 2
745:                                       add $t4, $t0, $t3
746: #                                     goto endI18;
747:                                       j endI18
748: #//                                }
749: else18:#//                            else
750: #//                                    {
751: #                                     //if (endPtr == a2 + used2)
752: #                                     if (endPtr != a2 + used2) goto else19;
753:                                       la $a0, a2
754:                                       sll $t0, $a2, 2
755:                                       add $t0, $t0, $a0
756:                                       bne $t4, $t0, else19
757: begI19:#//                            {
758: #                                             cout << comAeStr << 3 << comAfStr;
759:                                           li $v0, 4
760:                                           la $a0, comAeStr
761:                                           syscall
762:                                           li $v0, 1
763:                                           li $a0, 3
```

```
764:                                        syscall
765:                                        li $v0, 4
766:                                        la $a0, comAfStr
767:                                        syscall
768: #                                          hopPtr = a3;
769:                                        la $t3, a3
770: #                                          endPtr = hopPtr + used3;
771:                                        sll $t0, $a3, 2
772:                                        add $t4, $t0, $t3
773: #                                      goto endI19;
774:                                        j endI19
775: #//                                  }
776: else19:#//                          else
777: #//                                  {
778: #                                          //if (endPtr == a3 + used3)
779: #                                          if (endPtr != a3 + used3) goto endI20;
780:                                        la $a0, a3
781:                                        sll $t0, $a3, 2
782:                                        add $t0, $t0, $a0
783:                                        bne $t4, $t0, endI20
784: begI20:#//                                 {
785: #                                          cout << comAeStr << 4 << comAfStr;
786:                                        li $v0, 4
787:                                        la $a0, comAeStr
788:                                        syscall
789:                                        li $v0, 1
790:                                        li $a0, 4
791:                                        syscall
792:                                        li $v0, 4
793:                                        la $a0, comAfStr
794:                                        syscall
795: #                                          //if (used4 == 0)
796: #                                          if (used4 != 0) goto endI21;
797:                                        bnez $v1, endI21
798: begI21:#//                                 {
799: #                                              cout << endl;
800:                                            li $v0, 11
801:                                            li $a0, '\n'
802:                                            syscall
803: endI21:#//                                 }
804: #                                          hopPtr = a4;
805:                                        la $t3, a4
806: #                                          endPtr = hopPtr + used4;
807:                                        sll $t0, $v1, 2
808:                                        add $t4, $t0, $t3
809: endI20:#//                                 }
810: endI19:#//                          }
811: endI18:#//                          }
812: endW13:#//                 }
813: WTest13:#                 if (0 == 0) goto begW13;
814:                           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
```

```
866: #//                            }
867: else23:#//                     else
868: #//                                {
869: #                                       *hopPtr = *hopPtr2;
870:                                     lw $t0, 0($t2)
871:                                     sw $t0, 0($t3)
872: #                                   ++hopPtr2;
873:                                     addi $t2, $t2, 4
874: endI23:#//                          }
875: #                               ++hopPtr;
876:                                 addi $t3, $t3, 4
877: endW15:#//                      }
878: WTest15:#                   if (hopPtr1 >= endPtr1) goto xitW15;
879:                             bge $t1, $t9, xitW15
880: #                           if (hopPtr2 < endPtr2) goto begW15;
881:                             blt $t2, $t8, begW15
882: xitW15:#
883: #                           //while (hopPtr1 < endPtr1)
884: #                               goto WTest16;
885:                             j WTest16
886: begW16:#//                  {
887: #                                   *hopPtr = *hopPtr1;
888:                                 lw $t0, 0($t1)
889:                                 sw $t0, 0($t3)
890: #                                   ++hopPtr1;
891:                                 addi $t1, $t1, 4
892: #                               ++hopPtr;
893:                                 addi $t3, $t3, 4
894: endW16:#//                  }
895: WTest16:#                   if (hopPtr1 < endPtr1) goto begW16;
896:                             blt $t1, $t9, begW16
897: #
898: #                               //while (hopPtr2 < endPtr2)
899: #                               goto WTest17;
900:                             j WTest17
901: begW17:#//                      {
902: #                               *hopPtr = *hopPtr2;
903:                                 lw $t0, 0($t2)
904:                                 sw $t0, 0($t3)
905: #                               ++hopPtr2;
906:                                 addi $t2, $t2, 4
907: #                               ++hopPtr;
908:                                 addi $t3, $t3, 4
909: endW17:#//                      }
910: WTest17:#                       if (hopPtr2 < endPtr2) goto begW17;
911:                                 blt $t2, $t8, begW17
912: endI14:#//                 }
913: #                       cout << proA1Str;
914:                         li $v0, 4
915:                         la $a0, proA1Str
916:                         syscall
```

```
917:
918: #                       hopPtr = a1;
919:                         la $t3, a1
920: #                           endPtr = hopPtr + used1;
921:                         sll $t0, $a1, 2
922:                         add $t4, $t3, $t0
923: #
924: #                       //while (0 == 0)
925: #                       goto WTest18;
926:                         j WTest18
927: begW18:#//              {
928: #                           //if (hopPtr == a4 + used4 && endPtr == a4 + used4) break;
929: #                           if (hopPtr != a4 + used4) goto nbk24;
930:                             la $a0, a4
931:                             sll $t0, $v1, 2
932:                             add $t0, $t0, $a0
933:                             bne $t3, $t0, nbk24
934: #                           if (endPtr == a4 + used4) goto brk24;
935:                             la $a0, a4
936:                             sll $t0, $v1, 2
937:                             add $t0, $t0, $a0
938:                             beq $t4, $t0, brk24
939: nbk24:#
940: #                           //while (hopPtr < endPtr)
941: #                           goto WTest19;
942:                             j WTest19
943: begW19:#//                  {
944: #                                   cout << *hopPtr << ' ' << ' ';
945:                                 li $v0, 1
946:                                 lw $a0, 0($t3)
947:                                 syscall
948:                                 li $v0, 11
949:                                 li $a0, ' '
950:                                 syscall
951:                                 li $v0, 11
952:                                 li $a0, ' '
953:                                 syscall
954: #                                   ++hopPtr;
955:                                 addi $t3, $t3, 4
956: endW19:#//                  }
957: WTest19:#                   if (hopPtr < endPtr) goto begW19;
958:                             blt $t3, $t4, begW19
959: #                           cout << endl;
960:                             li $v0, 11
961:                             li $a0, '\n'
962:                             syscall
963: #                           //if (endPtr == a1 + used1)
964: #                           if (endPtr != a1 + used1) goto else25;
965:                             la $a0, a1
966:                             sll $t0, $a1, 2
967:                             add $t0, $t0, $a0
```

```
968:                            bne $t4, $t0, else25
969: begI25:#//                 {
970: #                                  cout << comAeStr << 2 << comAfStr;
971:                                li $v0, 4
972:                                la $a0, comAeStr
973:                                syscall
974:                                li $v0, 1
975:                                li $a0, 2
976:                                syscall
977:                                li $v0, 4
978:                                la $a0, comAfStr
979:                                syscall
980: #                                  hopPtr = a2;
981:                                la $t3, a2
982: #                                endPtr = hopPtr + used2;
983:                                sll $t0, $a2, 2
984:                                add $t4, $t0, $t3
985: #                                goto endI25;
986:                                j endI25
987: #//                            }
988: else25:#//                      else
989: #//                            {
990: #                                  //if (endPtr == a2 + used2)
991: #                                  if (endPtr != a2 + used2) goto else26;
992:                                la $a0, a2
993:                                sll $t0, $a2, 2
994:                                add $t0, $t0, $a0
995:                                bne $t4, $t0, else26
996: begI26:#//                      {
997: #                                  cout << comAeStr << 3 << comAfStr;
998:                                     li $v0, 4
999:                                     la $a0, comAeStr
1000:                                    syscall
1001:                                    li $v0, 1
1002:                                    li $a0, 3
1003:                                    syscall
1004:                                    li $v0, 4
1005:                                    la $a0, comAfStr
1006:                                    syscall
1007: #                                  hopPtr = a3;
1008:                                    la $t3, a3
1009: #                                  endPtr = hopPtr + used3;
1010:                                    sll $t0, $a3, 2
1011:                                    add $t4, $t0, $t3
1012: #                                      goto endI26;
1013:                                    j endI26
1014: #//                                }
1015: else26:#//                         else
1016: #//                               {
1017: #                                  //if (endPtr == a3 + used3)
1018: #                                  if (endPtr != a3 + used3) goto endI27;
```

```
1019:                                   la $a0, a3
1020:                                   sll $t0, $a3, 2
1021:                                   add $t0, $t0, $a0
1022:                                   bne $t4, $t0, endI27
1023: begI27:#//                        {
1024: #                                     cout << comAeStr << 4 << comAfStr;
1025:                                       li $v0, 4
1026:                                       la $a0, comAeStr
1027:                                       syscall
1028:                                       li $v0, 1
1029:                                       li $a0, 4
1030:                                       syscall
1031:                                       li $v0, 4
1032:                                       la $a0, comAfStr
1033:                                       syscall
1034: #                                     //if (used4 == 0)
1035: #                                     if (used4 != 0) goto endI28;
1036:                                       bnez $v1, endI28
1037: begI28:#//                                {
1038: #                                         cout << endl;
1039:                                           li $v0, 11
1040:                                           li $a0, '\n'
1041:                                           syscall
1042: endI28:#//                               }
1043: #                                         hopPtr = a4;
1044:                                       la $t3, a4
1045: #                                         endPtr = hopPtr + used4;
1046:                                       sll $t0, $v1, 2
1047:                                       add $t4, $t0, $t3
1048: endI27:#//                            }
1049: endI26:#//                        }
1050: endI25:#//                    }
1051: endW18:#//                    }
1052: WTest18:#                    if (0 == 0) goto begW18;
1053:                           j begW18
1054: brk24:#
1055: #
1056: #                          cout << endl;
1057:                           li $v0, 11
1058:                           li $a0, '\n'
1059:                           syscall
1060: #                          cout << dacStr;
1061:                           li $v0, 4
1062:                           la $a0, dacStr
1063:                           syscall
1064: #                          cin >> reply;
1065:                           li $v0, 12
1066:                           syscall
1067:                           move $t6, $v0
1068: #                          cout << endl;
1069:                           li $v0, 11
```

```
1070:                       li $a0, '\n'
1071:                       syscall
1072: endDW1:#      }
1073: DWTest1:#   //while (reply != 'n' && reply != 'N');
1074: #             if (reply == 'n') goto xitDW1;
1075:           li $t0, 'n'
1076:           beq $t6, $t0, xitDW1
1077: #             if (reply != 'N') goto begDW1;
1078:           li $t0, 'N'
1079:           bne $t6, $t0, begDW1
1080:
1081: xitDW1:
1082: #
1083: #             cout << dlStr;
1084: #         cout << '\n';
1085:           li $v0, 4
1086:           la $a0, dlStr
1087:           syscall
1088: #             cout << byeStr;
1089: #             cout << '\n';
1090:           li $v0, 4
1091:           la $a0, byeStr
1092:           syscall
1093: #         cout << dlStr;
1094: #         cout << '\n';
1095:           li $v0, 4
1096:           la $a0, dlStr
1097:           syscall
1098: #             return 0;
1099: #}
1100:           li $v0, 10      # graceful exit
1101:           syscall
```