```
################################################################################
# Title: Assign02P3                     Author: Clayton Stamper
# Class: CS 2318-004, Fall 2018         Submitted: 11/1/2018
################################################################################
# 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];
#char einStr[]    = "Enter integer #";
#char moStr[]     = "Max of ";
#char ieStr[]     = " ints entered...";
#char emiStr[]    = "Enter more ints? (n or N = no, others = yes) ";
#char begA1Str[]  = "beginning a1: ";
#char procA1Str[] = "processed a1: ";
#char commA2Str[] = "          a2: ";
#char commA3Str[] = "          a3: ";
#char dacStr[]    = "Do another case? (n or N = no, others = yes) ";
#char dlStr[]     = "================================\n";
#char byeStr[]    = "bye...";

                             .data
a1:                          .space, 48
a2:                          .space, 48
a3:                          .space, 48
einStr:                      .asciiz, "\nEnter integer #"
moStr:                       .asciiz, "Max of "
ieStr:                       .asciiz, " ints entered..."
emiStr:                      .asciiz, "\nEnter more ints? (n or N = no, others = yes) "
begA1Str:                    .asciiz, "beginning a1: "
procA1Str:                   .asciiz, "processed a1: "
commA2Str:                   .asciiz, "          a2: "
commA3Str:                   .asciiz, "          a3: "
dacStr:                      .asciiz,  "Do another case? (n or N = no, others = yes) "
dlStr:                       .asciiz,  "================================\n"
byeStr:                      .asciiz,  "bye...\n"

                             .text

                             ################################################
# Register usage:
#################
# $a1: endPtr1
# $a2: endPtr2
# $a3: endPtr3
# $t0: store-to address or temp-holder 2 (non-overlappingly)
# $t1: used1
# $t2: used2
# $t3: used3
# $t4: target
# $t5: hopPtr1
# $t6: hopPtr2
# $t7: hopPtr3
# $t8: hopPtr21 or mean (overlap with no harm)
# $t9: total
# $v1: reply or temp-holder 1 (non-overlappingly)
##################################################

#int main()
#{
#                char reply;
#                int used1,
#                    used2,
#                    used3,
#                    target,
#                   total,
#                    mean,
#                    *hopPtr1,
#                    *hopPtr2,
#                    *hopPtr21,
#                    *hopPtr3,
#                    *endPtr1,
#                    *endPtr2,
#                    *endPtr3;



#                cout << endl;
                             li $v0, 11
                             li $a0, '\n'
                             syscall
#                reply = 'y';
                             li $v1, 'y'
#                //while (reply != 'n' && reply != 'N')
#                goto WTest1;
                             j WTest1
begW1:#//        {
#                   used1 = 0;
#                   used2 = 0;
#                   used3 = 0;
                             li $t1, 0
                             li $t2, 0
                             li $t3, 0

#                   hopPtr1 = a1;
                             la $t5, a1

#                   //while (reply != 'n' && reply != 'N')
#                   goto WTest2;
                             j WTest2
begW2:#//           {
      #                cout << einStr;
                             li $v0, 4
                             la $a0, einStr
                             syscall
#                      cout << (used1 + 1);
                             addi $a0, $t1, 1
```

```
                                li $v0, 1
                                syscall
                        #cout << ':' << ' ';
                                li $v0, 11
                                li $a0, ':'
                                syscall
                                li $a0, ' '
                                syscall
#                        cin >> *hopPtr1;
                                li $v0, 5
                                syscall
                                sw $v0, 0($t5)
 #                       ++used1;
                                addi $t1, $t1, 1
#                        ++hopPtr1;
                                addi $t5, $t5, 4
#                        //if (used1 < 12)
#                        if (used1 >= 12) goto elseI1;
                                li $t0, 12
                                bge $t1, $t0, elseI1
begI1:#//                {
#                           cout << emiStr;
                                li $v0, 4
                                la $a0, emiStr
                                syscall
 #                          cin >> reply;
                                li $v0, 12
                                syscall
                                move $v1, $v0
#                        goto endI1;
                                j endI1
#//                      }
elseI1:#//                else
#//                      {
#                           cout << moStr << 12 << ieStr << endl;
                                li $v0, 4
                                la $a0, moStr
                                syscall
                                li $v0, 1
                                li $a0, 12
                                syscall
                                li $v0, 4
                                la $a0, ieStr
                                syscall
                                li $v0, 11
                                li $a0, '\n'
                                syscall
#                           reply = 'n';
                                li $v1, 'n'
endI1:#//                }
WTest2:#//            }
 #                    ////if (reply != 'n' && reply != 'N') goto begW2;
#                     if (reply == 'n') goto xitW2;
                                li $t0, 'n'
                                beq $v1, $t0, xitW2
#                     if (reply != 'N') goto begW2;
                                li $t0, 'N'
                                bne, $v1, $t0, begW2
```

```
xitW2:
#                  cout << endl;
                              li $v0, 11
                              li $a0, '\n'
                             syscall

#                  //if (used1 > 0)
#                  if (used1 <= 0) goto endI2;
                             ble $t1, $zero, endI2
begI2:#//          {
#                     total = 0;
                             li $t9, 0

 #                   //for (hopPtr1 = a1, endPtr1 = a1 + used1; hopPtr1 < endPtr1; ++hopPtr1
 #                    hopPtr1 = a1;
                             la $t5, a1
 #                    endPtr:ed1;
                             la $a1, a1
                             add $a1, $a1, $t1
#                     goto FTest1;
                             j FTest1
begF1:#//             {
#                        target = *hopPtr1;
                             lw $t4, 0($t5)
#                        total += target;
                             add $t9, $t9, $t4
 #                       //if (target % 2 == 1)
 #                       ////if (target % 2 != 1) goto elseI3;
#                        if ( (target & 1) != 1) goto elseI3;
                             andi $v1, $t4, 1
                             beqz $v1, elseI3
begI3:#//                {
#                           hopPtr3 = a3 + used3 - 1;
                             la $t7, a3
                             addi $t0, $t3, -1 #// hopPtr3 = hopPtr3 - 4
                             sll $t0, $t0, 2 #// hopPtr3 = used3^2
                             add $t7, $t7, $t0 #// hopPtr3 = hopPtr3 + a3

#                           endPtr3 = a3;
                             la $a3, a3
#                           //while (hopPtr3 >= endPtr3)
 #                          goto WTest3;
                             j WTest3
begW3:#//                   {
#                              //if (*hopPtr3 > target)
#                              if (*hopPtr3 <= target) goto elseI4;
                             lw $v1, 0($t5)
                             ble $v1, $t4, elseI4
begI4:#//                      {
 #                                *(hopPtr3 + 1) = *hopPtr3;
                             lw $v1, 0($t7)
                             sw $v1, 4($t7)

#                                 --hopPtr3;
                             addi $t7, $t7, -4

#                              goto endI4;
                             j endI4
```

```
#//                            }
elseI4:#//                      else
#//                            {
#                                 //break;
#                                 goto brk1;
                             j brk1
endI4:#//                      }
WTest3:#//                  }
#                           if (hopPtr3 >= endPtr3) goto begW3;
                             bge $t7, $a3, begW3
brk1:#

#                           *(hopPtr3 + 1) = target;
                             sw $t4, 4($t7)
#                           ++used3;
                             addi $t3, $t3, 1
#                        goto endI3;
                             j endI3
#//                      }
elseI3:#//                else
#//                      {
#                           hopPtr2 = a2;
                             la $t6, a2
#                           endPtr2 = a2 + used2;
                             sll $t0, $t2, 2
                             add $a2, $t6, $t0
#                           //while (hopPtr2 < endPtr2)
#                           goto WTest4;
                             j WTest4
begW4:#//                   {
#                              //if (*hopPtr2 >= target)
#                              if (*hopPtr2 < target) goto elseI5;
                             lw $v1, 0($t6)
                             blt $v1, $t4, elseI5
begI5:#//                      {
#                                 hopPtr21 = endPtr2;
                             move $t8, $t6
 #                                ///while (hopPtr21 > hopPtr2)
#                                 goto WTest5;
                             j WTest5
begW5:#//                         {
#                                    *hopPtr21 = *(hopPtr21 - 1);
                             lw $v1, -4($t8)
                             sw $v1 0($t8)
#                                    --hopPtr21;
                             li $t0, -4
                             add $t8, $t8, $t0
WTest5:#//                        }
#                                 if (hopPtr21 > hopPtr2) goto begW5;
                             bgt $t8, $t6, begW5

#                                 ///break;
#                                 goto brk2;
                             j brk2
#                           ///goto endI5; // unreacheable
#//                            }
elseI5:#//                      else
#//                            {
#                                 ++hopPtr2;
```

```
                              addi $6, $t6 4
endI5:#//                       }
WTest4:#//                     }
#                              if (hopPtr2 < endPtr2) goto begW4;
                                blt $t6, $a2, begW4
brk2:
#                              *hopPtr2 = target;
                                sw $t4, 0($t6)
#                              ++used2;
                                addi $t2, $t2, 1
endI3:#//                   }
#                           mean = total/used1;
                                div $t9, $t1
                                mflo $t7
 #                       ++hopPtr1;
                                addi $t5, $t5, 4
FTest1:#//               }
#                        if (hopPtr1 < endPtr1) goto begF1;
                                blt $t5, $a1, begF1

#                        cout << begA1Str;
                                li $v0, 4
                                la $a0, begA1Str
                                syscall
#                        //if (used1 > 0)
#                        if (used1 <= 0) goto endI6;
                                ble $t1, $zero, endI6
begI6:#//                {
#                           hopPtr1 = a1;
                                la $t5, a1
#                           endPtr1 = a1 + used1;
                                sll $t0, $t1, 2
                                add $a1, $t5, $t0
#//                         do
begDW1:#//                  {
#                              cout << *hopPtr1 << ' ' << ' ';
                                lw $v1, 0($t5)
                                li $v0, 1
                                move $a0, $v1
                                syscall
                                li $v0, 11
                                li $a0 ' '
                                syscall
                                li $v0, 11
                                li $a0 ' '
                                syscall

#                              ++hopPtr1;
                                addi $t5, $t5, 4
DWTest1:#//                 }
#                           //while (hopPtr1 < endPtr1);
 #                          if (hopPtr1 < endPtr1) goto begDW1;
                                blt $t5, $a1, begDW1
endI6:#//                }
#                        cout << endl;
                                li $v0, 11
                                li $a0, '\n'
                                syscall
```

```
#                     cout << commA2Str;
                             li $v0, 4
                             la $a0, commA2Str
                             syscall
#                     //if (used2 > 0)
#                     if (used2 <= 0) goto endI7;
                             ble $t2, $zero, endI7
begI7:#//             {
#                        hopPtr2 = a2;
                             la $t6, a2
#                        endPtr2 = a2 + used2;
                             sll $t0, $t2, 2
                             add $a2, $t6, $t0
#//                      do
begDW2:#//               {
#                           cout << *hopPtr2 << ' ' << ' ';
                             li $v0, 1
                             lw $a0, 0($t6)
                             syscall
                             li $v0, 11
                             li $a0 ' '
                             syscall
                             li $v0, 11
                             li $a0 ' '
                             syscall
#                           ++hopPtr2;
                             addi $t6, $t6, 4
DWTest2:#//              }
#                        //while (hopPtr2 < endPtr2);
 #                       if (hopPtr2 < endPtr2) goto begDW2;
                             blt $t6, $a2, begDW2
endI7:#//             }
#                     cout << endl;
                             li $v0, 11
                             li $a0, '\n'
                             syscall

#                     cout << commA3Str;
                             li $v0, 4
                             la $a0, commA3Str
                             syscall

#                     //if (used3 > 0)
#                     if (used3 <= 0) goto endI8;
                             ble $t3, $zero, endI8
begI8:#//             {
#                        hopPtr3 = a3;
                             la $t7, a3
#                        endPtr3 = a3 + used3;
                             sll $t0, $t3, 2
                             add $a3, $t7, $t0
#//                      do
begDW3:#//               {
#                           cout << *hopPtr3 << ' ' << ' ';
                             li $v0, 1
                             lw $a0, 0($t7)
                             syscall
```

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

#                              ++hopPtr3;
                                addi $t7, $t7, 4
DWTest3:#//                 }
#                           //while (hopPtr3 < endPtr3);
#                           if (hopPtr3 < endPtr3) goto begDW3;
                                blt $t7, $a3, begDW3
endI8:#//                }
#                        cout << endl;
                                li $v0, 11
                                li $a0, '\n'
                                syscall
 #                       hopPtr1 = a1;
                                la $t5, a1
#                        hopPtr2 = a2;
                                la $t6, a2
#                        hopPtr3 = a3;
                                la $t7, a3
#                        endPtr2 = a2 + used2;
                                sll $t0, $t2, 2
                                add $a2, $t6, $t0
#                        endPtr3 = a3 + used3;
                                sll $t0, $t3, 2
                                add $a3, $t7, $t0
 #                       //while (hopPtr2 < endPtr2 && hopPtr3 < endPtr3)
#                        goto WTest6;
                                j WTest6
begW6:#//                {
#                           //if (*hopPtr2 < *hopPtr3)
#                           if (*hopPtr2 >= *hopPtr3) goto elseI9;
                                lw $v1, 0($t6)
                                lw $t0, 0($t7)
                                bge $v1, $t0, elseI9
begI9:#//                   {
#                              *hopPtr1 = *hopPtr2;
                                lw $v1, 0($t6)
                                sw $v1, 0($t5)
#                              ++hopPtr2;
                                addi $t6, $t6, 4
#                           goto endI9;
                                j endI9
#//                         }
elseI9:#//                   else
#//                         {
#                              *hopPtr1 = *hopPtr3;
                                lw $v1, 0($t7)
                                sw $v1, 0($t5)
#                              ++hopPtr3;
                                addi $t7, $t7, 4
endI9:#//                   }
#                           ++hopPtr1;
                                addi $t5, $t5, 4
```

```
WTest6:#//             }
#                      ////if (hopPtr2 < endPtr2 && hopPtr3 < endPtr3) goto begW6;
#                      if (hopPtr2 >= endPtr2) goto xitW6;
                              bge $t6, $12, xitW6
#                      if (hopPtr3 < endPtr3) goto begW6;
                              blt $t7, $a3, begW6
xitW6:
#                      //while (hopPtr2 < endPtr2)
#                     goto WTest7;
                              j WTest7
begW7:#//              {
#                         *hopPtr1 = *hopPtr2;
                              lw $v1, 0($t6)
                              sw $v1, 0($t5)
#                         ++hopPtr2;
                              addi $t6, $t6, 4
#                         ++hopPtr1;
                              addi $t5, $t5, 4
WTest7:#//             }
#                      if (hopPtr2 < endPtr2) goto begW7;
                              blt $t6, $a2, begW7

#                      //while (hopPtr3 < endPtr3)
#                      goto WTest8;
                              j WTest8
begW8:#//              {
#                         *hopPtr1 = *hopPtr3;
                              lw $v1, 0($t7)
                              sw $v1, 0($t5)
#                         ++hopPtr3;
                              addi $t7, $t7, 4
#                         ++hopPtr1;
                              addi $t5, $t5, 4
WTest8:#//             }
#                      if (hopPtr3 < endPtr3) goto begW8;
                              blt $t7, $a3, begW8

#                      hopPtr2 = a2;
                              la $t6, a2
#                      hopPtr3 = a3;
                              la $t7, a3
#                      used2 = 0;
                              li $t2, 0
#                      used3 = 0;
                              li $t3, 0
#                      //for (hopPtr1 = a1, endPtr1 = a1 + used1; hopPtr1 < endPtr1; ++hopPt
#                      hopPtr1 = a1;
                              la $t5, a1
#                      endPtr1 = a1 + used1;
                              sll $t0, $t1, 2
                              add $a1, $t5, $t0
#                      goto FTest2;
                              j FTest2
begF2:#//              {
#                         target = *hopPtr1;
                              lw $t4, 0($t5)
#                         //if (target < mean)
#                         if (target >= mean) goto elseI10;
```

```
                               bge $t4, $t8, elseI10
begI10:#//                {
#                            *hopPtr2 = target;
                               sw $t4, 0($t6)
#                            ++used2;
                               addi $t2, $t2, 1
#                            ++hopPtr2;
                               addi $t6, $t6, 4
#                         goto endI10;
                               j endI10
#//                       }
elseI10:#//                else
#//                       {
 #                           //if (target > mean)
 #                           if (target <= mean) goto endI11;
                               ble $t4, $t8, endI11
begI11:#//                   {
#                               *hopPtr3 = target;
                               sw $t4, 0($t7)
#                               ++used3;
                               addi $t3, $t3, 1
#                               ++hopPtr3;
                               addi $t7, $t7, 4
endI11:#//                   }
endI10:#//                }
#                      ++hopPtr1;
                               addi $t5, $t5, 4
FTest2:#//             }
#                      if (hopPtr1 < endPtr1) goto begF2;
                               blt $t5, $a1, begF2

#                      cout << procA1Str;
                               li $v0, 4
                               la $a0, procA1Str
                               syscall
#                      //if (used1 > 0)
#                      if (used1 <= 0) goto endI12;
                               ble $t1, $zero, endI12
begI12:#//             {
#                         hopPtr1 = a1;
                               la $t5, a1
#                         endPtr1 = a1 + used1;
                               sll $t0, $t1, 2
                               add $a1, $t5, $t0
#                         //do
begDW4:#//                {
#                            cout << *hopPtr1 << ' ' << ' ';
                               li $v0, 1
                               lw $a0, 0($t5)
                               syscall
                               li $v0, 11
                               li $a0, ' '
                               syscall
                               li $v0, 11
                               li $a0, ' '
                               syscall

#                            ++hopPtr1;
```

```
                                  addi $t5, $t5, 4
DWTest4:#//                     }
#                               //while (hopPtr1 < endPtr1);
#                               if (hopPtr1 < endPtr1) goto begDW4;
                                  blt $t5, $a1, begDW4
endI12:#//                   }
#                            cout << endl;
                                  li $v0, 11
                                  li $a0, '\n'
                                  syscall

#                            cout << commA2Str;
                                  li $v0, 4
                                  la $a0, commA2Str
                                  syscall
#                            //if (used2 > 0)
#                            if (used2 <= 0) goto endI13;
                                  ble $t2, $zero, endI13

begI13:#//                   {
#                               hopPtr2 = a2;
                                  la $t6, a2
#                               endPtr2 = a2 + used2;
                                  sll $t0, $t2, 2
                                  add $a2, $t6, $t0
#                               //do
begDW5:#//                      {
#                                  cout << *hopPtr2 << ' ' << ' ';
                                  li $v0, 1
                                  lw $a0, 0($t6)
                                  syscall
#                                 ++hopPtr2;
                                  addi $t6, $t6, 4
DWTest5:#//                     }
#                               //while (hopPtr2 < endPtr2);
#                               if (hopPtr2 < endPtr2) goto begDW5;
                                  blt $t6, $a2 begDW5
endI13:#//                   }
#                            cout << endl;
                                  li $v0, 11
                                  li $a0, '\n'
                                  syscall

#                            cout << commA3Str;
                                  li $v0, 4
                                  la $a0, commA3Str
                                  syscall
#                            //if (used3 > 0)
#                            if (used3 <= 0) goto endI14;
                                  ble $t3, $zero, endI14
begI14:#//                   {
#                               hopPtr3 = a3;
                                  la $t7, a3
#                               endPtr3 = a3 + used3;
                                  sll $t0, $t3, 2
                                  add $a3, $t7, $t0
#                               //do
begDW6:#//                      {
     #                           cout << *hopPtr3 << ' ' << ' ';
                                  li $v0, 1
                                  lw $a0, 0($t7)
                                  syscall
                                  li $v0, 11
                                  li $a0, ' '
                                  syscall
                                  li $v0, 11
                                  li $a0, ' '
                                  syscall
#                            ++hopPtr3;
                                  addi $t7, $t7, 4
DWTest6:#//                 }
#                        //while (hopPtr3 < endPtr3);
#                        if (hopPtr3 < endPtr3) goto begDW6;
                                  blt $t7, $a3, begDW6
endI14:#//              }
#                       cout << endl;
                                  li $v0, 11
                                  li $a0, '\n'
                                  syscall
endI2:#//            }

#                    cout << endl;
                                  syscall
#                    cout << dacStr;
                                  li $v0, 4
                                  la $a0, dacStr
                                  syscall
#                    cin >> reply;
                                  li $v0, 12
                                  syscall
                                  move $v1, $v0
#                    cout << endl;
                                  li $v0, 11
                                  li $a0, '\n'
                                  syscall
WTest1:#//        }
#                 ////if (reply != 'n' && reply != 'N') goto begW1;
#                 if (reply == 'n') goto xitW1;
                                  li $t0, 'n'
                                  beq $v1, $t0, xitW1
#                 if (reply != 'N') goto begW1;
                                  li $t0, 'N'
                                  bne, $v1, $t0, begW1
xitW1:
#                 cout << dlStr << '\n';
                                  li $v0, 4
                                  la $a0, dlStr
                                  syscall
#                 cout << byeStr << '\n';
                                  la $a0, byeStr
                                  syscall
#                 cout << dlStr << '\n';
                                  la $a0, dlStr
                                  syscall

#                 return 0;
```

```
                              li $v0, 10
                              syscall

#}
output:


Enter integer #1: 4

Enter more ints? (n or N = no, others = yes) n
beginning a1: 4
          a2: 4
          a3: 
processed a1: 4
          a2: 
          a3: 4

Do another case? (n or N = no, others = yes)


Enter integer #1: y


Enter integer #1: 4

Enter more ints? (n or N = no, others = yes) n
beginning a1: 4
          a2: 4
          a3: 
processed a1: 4
          a2: 
          a3: 4

Do another case? (n or N = no, others = yes) y

Enter integer #1: 5

Enter more ints? (n or N = no, others = yes) n
beginning a1: 5
          a2: 
          a3: 5
processed a1: 5
          a2: 
          a3: 5

Do another case? (n or N = no, others = yes) y

Enter integer #1: 4

Enter more ints? (n or N = no, others = yes) y
Enter integer #2: 5

Enter more ints? (n or N = no, others = yes) n
beginning a1: 4  5
          a2: 4
          a3: 
processed a1: 4  5
          a2: 
```

```
          a3: 4  5

Do another case? (n or N = no, others = yes) y

Enter integer #1: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #2: 3

Enter more ints? (n or N = no, others = yes) n
beginning a1: 5  3
          a2:
          a3: 5
processed a1: 5  3
          a2:
          a3: 5  3

Do another case? (n or N = no, others = yes) y

Enter integer #1: 8

Enter more ints? (n or N = no, others = yes) y
Enter integer #2: 6

Enter more ints? (n or N = no, others = yes) y
Enter integer #3: 3

Enter more ints? (n or N = no, others = yes) y
Enter integer #4: 4

Enter more ints? (n or N = no, others = yes) n
beginning a1: 8  6  3  4
          a2: 8
          a3:
processed a1: 8  6  3  4
          a2:
          a3: 8  6  3  4

Do another case? (n or N = no, others = yes) y

Enter integer #1: 1

Enter more ints? (n or N = no, others = yes) y
Enter integer #2: 3

Enter more ints? (n or N = no, others = yes) y
Enter integer #3: 4

Enter more ints? (n or N = no, others = yes) y
Enter integer #4: 6

Enter more ints? (n or N = no, others = yes) y
Enter integer #5: 7

Enter more ints? (n or N = no, others = yes) n
beginning a1: 1  3  4  6  7
          a2:
          a3: 1  3
```

```
processed a1: 1  3  4  6  7
          a2:
          a3: 1  3  4  6  7

Do another case? (n or N = no, others = yes) y

Enter integer #1: 4

Enter more ints? (n or N = no, others = yes) y
Enter integer #2: 4

Enter more ints? (n or N = no, others = yes) y
Enter integer #3: 4

Enter more ints? (n or N = no, others = yes) y
Enter integer #4: 4

Enter more ints? (n or N = no, others = yes) n
beginning a1: 4  4  4  4
          a2: 4
          a3:
processed a1: 4  4  4  4
          a2:
          a3: 4  4  4  4

Do another case? (n or N = no, others = yes) y

Enter integer #1: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #2: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #3: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #4: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #5: 5

Enter more ints? (n or N = no, others = yes) n
beginning a1: 5  5  5  5  5
          a2:
          a3: 5  5
processed a1: 5  5  5  5  5
          a2:
          a3: 5  5  5  5  5

Do another case? (n or N = no, others = yes) y

Enter integer #1: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #2: 2

Enter more ints? (n or N = no, others = yes) y
Enter integer #3: 9
Enter more ints? (n or N = no, others = yes) y
Enter integer #4: 0

Enter more ints? (n or N = no, others = yes) y
Enter integer #5: 7

Enter more ints? (n or N = no, others = yes) y
Enter integer #6: 1

Enter more ints? (n or N = no, others = yes) y
Enter integer #7: 3

Enter more ints? (n or N = no, others = yes) y
Enter integer #8: 6

Enter more ints? (n or N = no, others = yes) y
Enter integer #9: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #10: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #11: 8

Enter more ints? (n or N = no, others = yes) y
Enter integer #12: 2
Max of 12 ints entered...

beginning a1: 5  2  9  0  7  1  3  6  5  5  8  2
          a2: 2
          a3: 5  9
processed a1: 2  5  9  0  7  1  3  6  5  5  8  2
          a2:
          a3: 2  5  9  7  1  3  6  5  5  8  2

Do another case? (n or N = no, others = yes) y

Enter integer #1: 5

Enter more ints? (n or N = no, others = yes) y
Enter integer #2: 2

Enter more ints? (n or N = no, others = yes) y
Enter integer #3: 9

Enter more ints? (n or N = no, others = yes) y
Enter integer #4: 0

Enter more ints? (n or N = no, others = yes) y
Enter integer #5: 7

Enter more ints? (n or N = no, others = yes) y
Enter integer #6: 1

Enter more ints? (n or N = no, others = yes) y
Enter integer #7: 3
Enter more ints? (n or N = no, others = yes) y
Enter integer #8: 6

Enter more ints? (n or N = no, others = yes) y
Enter integer #9: 4

Enter more ints? (n or N = no, others = yes) y
Enter integer #10: 4

Enter more ints? (n or N = no, others = yes) y
Enter integer #11: 8

Enter more ints? (n or N = no, others = yes) y
Enter integer #12: 2
Max of 12 ints entered...

beginning a1: 5  2  9  0  7  1  3  6  4  4  8  2
          a2: 2
          a3: 5  9
processed a1: 2  5  9  0  7  1  3  6  4  4  8  2
          a2:
          a3: 2  5  9  7  1  3  6  4  4  8  2

Do another case? (n or N = no, others = yes) n
================================
bye...
================================

-- program is finished running --
```

L)

#######

####

####

####

# # #