```
                        ;Lab6 CSCI 112
                        ;Bee Cha
                        ;Prompt the user to enter their name
                        ;Prompt the user to enter N number of test scores (N = 1~10)
                        ;If N is out of range, then throw an error to reenter N
                        ;Test scores range (0~100)
                        ;Purpose of the lab, N=8
                        ;Scores: 76, 88, 95, 67, 59, 84, 95, 77
                        ;Using loop instructions, store each test score into an array
                        ;Using loop instructions, calculate total store and average
                        ;Then output the results in one box


                        .586
                        .MODEL FLAT

                        INCLUDE io.h            ; header file for input/output
                       C ; IO.H -- header file for I/O macros (listing suppressed)
                       C .NOLIST      ; turn off listing
                       C .LIST        ; begin listing
                       C

                        .STACK 8192

 00000000               .DATA

 00000000 53 74 75 64 65  outlbl BYTE   'Student Record', 0
          6E 74 20 52 65
          63 6F 72 64 00
 0000000F 52 65 73 75 6C  label1 BYTE   'Results for ', 0                        ;12
characters
          74 73 20 66 6F
          72 20 00
 0000001C 54 6F 74 61 6C  label2 BYTE   'Total Score: ', 0                       ;13
characters
          20 53 63 6F 72
          65 3A 20 00
 0000002A 54 6F 74 61 6C  label3 BYTE   'Total Average: ', 0              ;15 characters
          20 41 76 65 72
          61 67 65 3A 20
          00
 0000003A 45 6E 74 65 72  prompt1       BYTE    "Enter your name", 0
          20 79 6F 75 72
          20 6E 61 6D 65
          00
 0000004A 45 6E 74 65 72  prompt2       BYTE    "Enter number of test score", 0
          20 6E 75 6D 62
          65 72 20 6F 66
          20 74 65 73 74
          20 73 63 6F 72
          65 00
 00000065 45 6E 74 65 72  prompt3       BYTE   "Enter Scores", 0
          20 53 63 6F 72
          65 73 00
 00000072 45 52 52 4F 52  errlbl BYTE   'ERROR', 0
          00
```

```
00000078 49 6E 63 6F 72    error1 BYTE   'Incorrrect range. Please enter 1~10', 0
         72 72 65 63 74
         20 72 61 6E 67
         65 2E 20 50 6C
         65 61 73 65 20
         65 6E 74 65 72
         20 31 7E 31 30
         00
0000009C  00000028 [             string  BYTE    40 DUP (?)
          00
         ]
000000C4  00000190 [             outstr BYTE   400 DUP (?)
          00
         ]
00000254  0000000A [             t_score       DWORD  10 DUP (0)
          00000000
         ]
0000027C 00000000          numele DWORD  ?
00000280 00000000          sum           DWORD  ?
00000284  0000000B [             sumstr BYTE   11 DUP (?)
          00
         ]
0000028F 00000000          avg           DWORD  ?
00000293  0000000B [             avgstr BYTE   11 DUP (?)
          00
         ]

00000000                   .CODE
00000000                   _MainProc PROC


                                 ;-----Prepare string for output-----;

00000000  8D 35 0000000F R             lea           esi, label1
      ;12 characters
00000006  8D 3D 000000C4 R             lea           edi, outstr
0000000C  FC                           cld
0000000D  B9 0000000C                         mov           ecx, 12
00000012  F3/ A4                       rep           movsb

                                 ;-----Get User's name-----;

                                       input  prompt1, string, 40               ;Tell user
to enter name
                                       ;output       outlbl, string

                                 ;-----Append name to output string-----;

00000032  8D 35 0000009C R             lea           esi, string
00000038  8D 3D 000000D0 R             lea           edi, outstr+12
0000003E  FC                           cld
0000003F  B9 00000003                         mov           ecx, 3
00000044  F3/ A4                       rep           movsb
00000046  C6 05 000000D3 R             mov           outstr+15, 0dh
         0D
                                       ;output       outlbl, outstr

0000004D  EB 19                        jmp           INPUT_N
             ;initial skip of the error label
```

```
                                ;-----Prompt user to enter N-----;

 0000004F                   REENTER:
                                        output errlbl, error1

 00000068                   INPUT_N:
                                        input  prompt2, string, 4
                                        atod   string
      ;convert ASCII to DWORD
 00000095  A3 0000027C R                mov           numele, eax

                                ;-----Compare if N is less than 1-----;
                                ;-----EAX should contain the DWORD value of N-----;

 0000009A  83 F8 01                     cmp           eax, 1
 0000009D  7C B0                        jl            REENTER

                                ;-----Compare if N is greater than 10-----;
 0000009F  83 F8 0A                     cmp           eax, 10
 000000A2  7F AB                        jg            REENTER

                                ;-----Using loop to prompt user to enter N number of scores
into array-----;
                                ;While i < N
                                ;      store number into array[i]
                                ;      i++
                                ;End While
                                ;
                                ;In assembly's case, it'll be array[j] where j is
incremented by 4
                                ;array[j]
                                ;j+=4

 000000A4  8D 1D 00000254 R             lea           ebx, t_score               ;starting
address of array
 000000AA  8B C8                        mov           ecx, eax                         ;Move
N to ecx

 000000AC                   IN_ARRAY:
                                        input  prompt3, string, 10
                                        atod   string
 000000D9  89 03                        mov           DWORD PTR [ebx], eax
 000000DB  83 C3 04                     add           ebx, 4
 000000DE  E2 CC                        loop   IN_ARRAY

                                ;-----Calculate total average using loop instruction-----;
                                ;-----Total Score and Average-----;
                                ;76+88+95+67+59+84+95+77 = 641 or 281h
                                ;641/8 = 80.125 or 50h

 000000E0  8D 1D 00000254 R             lea           ebx, t_score
 000000E6  8B 0D 0000027C R             mov           ecx, numele
 000000EC  B8 00000000                         mov           eax, 0
      ;start sum = 0

 000000F1                   AVG_LOOP:
 000000F1  03 03                        add           eax, [ebx]
 000000F3  83 C3 04                     add           ebx, 4
 000000F6  E2 F9                        loop   AVG_LOOP
 000000F8  A3 00000280 R                mov           sum, eax
```

```
 000000FD  F7 35 0000027C R              div           numele
      ;total divide of number of elements
 00000103  A3 0000028F R                 mov           avg, eax

                                ;-----Append results into string-----;

 00000108  8D 35 0000001C R              lea           esi, label2
 0000010E  8D 3D 000000D4 R              lea           edi, outstr+16             ;13
characters
 00000114  FC                            cld
 00000115  B9 0000000D                          mov           ecx, 13
 0000011A  F3/ A4                        rep           movsb


 0000011C  A1 00000280 R                 mov           eax, sum
                                         dtoa   sumstr, eax
 00000139  8D 35 00000284 R              lea           esi, sumstr
 0000013F  8D 3D 000000E1 R              lea           edi, outstr+29
 00000145  FC                            cld
 00000146  B9 0000000B                          mov           ecx, 11
 0000014B  F3/ A4                        rep           movsb
 0000014D  C6 05 000000EC R              mov           outstr+40, 0dh
          0D

 00000154  8D 35 0000002A R              lea           esi, label3
 0000015A  8D 3D 000000ED R              lea           edi, outstr+41             ;15
characters
 00000160  FC                            cld
 00000161  B9 0000000F                          mov           ecx, 15
 00000166  F3/ A4                        rep           movsb

 00000168  A1 0000028F R                 mov           eax, avg
                                         dtoa   avgstr, eax
 00000185  8D 35 00000293 R              lea           esi, avgstr
 0000018B  8D 3D 000000FC R              lea           edi, outstr+56
 00000191  FC                            cld
 00000192  B9 0000000B                          mov           ecx, 11
 00000197  F3/ A4                        rep           movsb

                                         output outlbl, outstr

 000001B2  B8 00000000                    mov     eax, 0  ; exit with return code 0
 000001B7  C3                        ret
 000001B8                    _MainProc ENDP
                             END                              ; end of source code

Microsoft (R) Macro Assembler Version 12.00.30501.0        03/06/15 14:12:58
lab6.asm                                                 Symbols 2 - 1



Macros:

                N a m e                 Type

atod . . . . . . . . . . . . . .  Proc
atow . . . . . . . . . . . . . .  Proc
dtoa . . . . . . . . . . . . . .  Proc
```

```
input  . . . . . .  Proc
output . . . . . .  Proc
wtoa . . . . . . .  Proc


Segments and Groups:

                N a m e                 Size     Length   Align   Combine Class

FLAT . . . . . . .  GROUP
STACK  . . . . . .  32 Bit  00002000 Para        Stack         'STACK'
_DATA  . . . . . .  32 Bit  0000029E Para        Public  'DATA'
_TEXT  . . . . . .  32 Bit  000001B8 Para        Public  'CODE'


Procedures, parameters, and locals:

                N a m e                 Type     Value    Attr

_MainProc  . . . . . .  P Near  00000000 _TEXT      Length= 000001B8 Public
  REENTER  . . . . . .  L Near  0000004F _TEXT
  INPUT_N  . . . . . .  L Near  00000068 _TEXT
  IN_ARRAY . . . . . .  L Near  000000AC _TEXT
  AVG_LOOP . . . . . .  L Near  000000F1 _TEXT


Symbols:

                N a m e                 Type     Value    Attr

@CodeSize  . . . . . .  Number  00000000h
@DataSize  . . . . . .  Number  00000000h
@Interface . . . . . .  Number  00000000h
@Model . . . . . . . .  Number  00000007h
@code  . . . . . . . .  Text           _TEXT
@data  . . . . . . . .  Text           FLAT
@fardata?  . . . . . .  Text           FLAT
@fardata . . . . . . .  Text           FLAT
@stack . . . . . . . .  Text           FLAT
_getInput  . . . . . .  L Near  00000000 FLAT       External
_showOutput  . . . . .  L Near  00000000 FLAT       External
atodproc . . . . . . .  L Near  00000000 FLAT       External
atowproc . . . . . . .  L Near  00000000 FLAT       External
avgstr . . . . . . . .  Byte    00000293 _DATA
avg  . . . . . . . . .  DWord   0000028F _DATA
dtoaproc . . . . . . .  L Near  00000000 FLAT       External
errlbl . . . . . . . .  Byte    00000072 _DATA
error1 . . . . . . . .  Byte    00000078 _DATA
label1 . . . . . . . .  Byte    0000000F _DATA
label2 . . . . . . . .  Byte    0000001C _DATA
label3 . . . . . . . .  Byte    0000002A _DATA
numele . . . . . . . .  DWord   0000027C _DATA
outlbl . . . . . . . .  Byte    00000000 _DATA
outstr . . . . . . . .  Byte    000000C4 _DATA
prompt1  . . . . . . .  Byte    0000003A _DATA
prompt2  . . . . . . .  Byte    0000004A _DATA
prompt3  . . . . . . .  Byte    00000065 _DATA
string . . . . . . . .  Byte    0000009C _DATA
sumstr . . . . . . . .  Byte    00000284 _DATA
sum  . . . . . . . . .  DWord   00000280 _DATA
```

```
t_score  . . . . . .  DWord   00000254 _DATA
wtoaproc . . . . . .  L Near  00000000 FLAT      External

          0 Warnings
          0 Errors
```