**Program I Binary**

/\*\* ASSUMED ADDRESSES FOR EACH PROCEDURE

\* encode proc 0x6000 [address: 00000000000000000110000000000000]

\* getline proc 0x7000 [address: 00000000000000000111000000000000]

\* WriteString 0x8000 [address: 00000000000000001000000000000000]

\* GetString 0x9000 [address: 00000000000000001001000000000000]

\* NewLine 0x10000 [address: 00000000000000010000000000000000]

\*/

/\*\* .DATA (showing assumed addresses) \*/

plain BYTE ?

0x1000 [address: 00000000000000000001000000000000]

code BYTE ?

0x2000 [address: 00000000000000000010000000000000]

prompt BYTE "Enter a string: ", 0

0x3000 [address: 00000000000000000011000000000000]

msg BYTE "The encoded string is:", 0

0x4000 [address: 00000000000000000100000000000000]

buff BYTE ?100

0x5000 [address: 00000000000000000101000000000000]

/\*\* END .DATA \*/

/\*\* ENCODE PROC \*/

compare\_ge plain, 'a'

010001 001 00000000000000000001000000000000 100 00000000000000000000000001100001

compare\_le plain 'x'

010010 011 00000000000000000001000000000000 100 00000000000000000000000001111000

branch\_and cond1\_true

010100 00000000000000000000000000000101

compare\_ge plain, 'A'

010001 001 00000000000000000001000000000000 100 00000000000000000000000001000001

compare\_le plain 'X'

010010 001 00000000000000000001000000000000 100 00000000000000000000000001011000

branch\_and cond1\_true

010100 00000000000000000000000000000010

jumpto else\_L

001001 00000000000000000000000000000011

/\*\* cond1\_true label \*/

add plain, 2

000000 001 00000000000000000001000000000000 100 00000010

jumpto end\_proc

001001 00000000000000000000000000001000

/\*\* cond2\_else label \*/

subtract plain, 24

000010 001 00000000000000000001000000000000 100 00011000

/\*\* end\_proc label \*/

move plain, code

001000 001 00000000000000000001000000000000 001 00000000000000000010000000000000

move code, RV1

001000 001 00000000000000000010000000000000 000 1000

return

010110

/\*\* END ENCODE PROC \*/

/\*\* GETLINE PROC \*/

move 0, GP1

001000 100 00000000 000 0000

/\*\* l label \*/

move [AR3+GP1],[AR1+GP1]

001000 011 [address at AR3+GP1] 011 [address at AR1+GP1]

add GP1, 1

000000 000 0000 100 00000001

compare\_lt GP1, AR2

010000 000 0000 000 0101

compare\_eq 0, [AR3+GP1]

010011 100 00000000 011 [address at [AR3+GP1]]

branch e

010101 00000000000000000000000000000010

jumpto l

001001 11111111111111111111111111111011

/\*\* e label \*/

return

010110

/\*\* end getline proc \*/

/\*\* .run \*/

move [prompt], AR1

001000 011 00000000000000000011000000000000 000 0100

call WriteString

001010 00000000000000001000000000000000

call GetString

001010 00000000000000001001000000000000

move [buff], AR1

001000 011 00000000000000000101000000000000 000 0100

move 100, AR2

001000 100 01100100 000 0101

move RV1, AR3

001000 000 1000 000 0110

call getline

001010 00000000000000000111000000000000

move 0, GP1

001000 100 00000000 000 0000

move buff[GP1], plain

001000 010 [value of buff[GP1]] 001 00000000000000000001000000000000

call encode

001010 00000000000000000110000000000000

move RV1, buff[GP1]

001000 000 1000 010 [value of buff[GP1]]

compare\_eq 0, buff[GP1]

010011 100 00000000 010 [value of buff[GP1]]

branch end\_of\_loop

010101 00000000000000000000000000000011

add GP1, 1

000000 000 0000 100 00000001

jumpto top\_loop

001001 11111111111111111111111111111010

/\*\* end\_of\_loop label \*/

move [msg], AR1

001000 011 00000000000000000100000000000000 000 0100

call WriteString

001010 00000000000000001000000000000000

move [buff], AR1

001000 011 00000000000000000101000000000000 000 0100

call WriteString

001010 00000000000000001000000000000000

call NewLine

001010 00000000000000010000000000000000

**Program II Binary**

/\*\* ASSUMED ADDRESSES FOR EACH PROCEDURE

\* WriteString 0x5000 [address: 00000000000000000101000000000000]

\* GetInt 0x6000 [address: 00000000000000000110000000000000]

\* WriteNewline 0x7000 [address: 00000000000000000111000000000000]

\* WriteInt 0x8000 [address: 00000000000000000001111101000000]

\*/

/\*\* .DATA (showing assumed addresses) \*/

sum DWORD ?

0x1000 [address: 00000000000000000001000000000000]

average DWORD ?

0x2000 [address: 00000000000000000010000000000000]

prompt BYTE "Enter a non-negative integer: ",0

0x3000 [address: 00000000000000000011000000000000]

message BYTE "Average= ",0

0x4000 [address: 00000000000000000100000000000000]

/\*\* end .data \*/

/\*\* .run \*/

move 0, GP4

001000 100 00000000000000000000000000000000 000 0011

move [prompt], AR1

001000 011 00000000000000000011000000000000 000 0100

call WriteString

001010 00000000000000000101000000000000

call GetInt

001010 00000000000000000110000000000000

add sum, RV1

000000 001 00000000000000000001000000000000 000 1000

call WriteNewline

001010 00000000000000000111000000000000

add GP4,1

000000 000 0011 100 00000001

compare\_lt GP4, 9

010000 000 0011 100 000001001

branch top\_loop

010101 11111111111111111111111111111001

divide2 sum, 10, average

000111 001 00000000000000000001000000000000 100 00001010 001 00000000000000000010000000000000

move [message], AR1

001000 011 00000000000000000100000000000000 000 0100

call WriteString

001010 00000000000000000101000000000000

move [average], AR1

001000 011 00000000000000000010000000000000 000 0100

call WriteInt

001010 00000000000000000001111101000000

call WriteNewline

001010 00000000000000000111000000000000

/\*\* end .run \*/