Introduction to HLASM – SHARE Boston 2013
High Level Assembler Bootcamp

Example 1

```
**********************************************************************
* SIMPLE HELLO WORLD PROGRAM
* Copyright IBM UK LTD 2013
**********************************************************************
*
* MAIN PROGRAM STARTS HERE
*
EX1      CSECT
EX1      AMODE 31
EX1      RMODE 24
 * USUAL PROGRAM SETUP    <- FIX THIS COMMENT
         STM   14,12,12(13)
         BALR  12,0
         USING *,12
*
* *********************************************************************
* WRITE YOUR CODE HERE
* MOVE THE DATA IN_STRING TO OUT_STRING
* HERE...
* *********************************************************************
*
         LA    5,WTO_AR
         WTO   TEXT=(5)
LMRET    LM    14,12,12(13)
*
* *********************************************************************
* WRITE YOUR CODE HERE
* THE RETURN CODE OF THE PROGRAM IS HANDED BACK IN REGISTER 15
* PROVIDE A RETURN CODE OF 15
* HERE...
* *********************************************************************
*
         BR    14
* *********************************************************************
* END OF PROGRAM
* *********************************************************************
IN_STRING  DC    C'HELLO WORLD!'
WTO_AR     DC    AL2(L'OUT_STRING)
OUT_STRING DS    CL(L'IN_STRING)
         LTORG ,
         END
```

Example 2

```
**********************************************************************
* SIMPLE ADDRESSING LOOP PROGRAM
* Copyright IBM UK LTD 2013
**********************************************************************
*
* MAIN PROGRAM STARTS HERE
*
EX2      CSECT
```

```
EX2      AMODE 31
EX2      RMODE 24
* USUAL PROGRAM SETUP
         STM   14,12,12(13)
         BALR  12,0
         USING *,12
*
* SAVE REGISTER 1 SOMEWHERE BECAUSE IT MAY BE USED BY WTO
* HERE...
*
         WTO   'HELLO'
         LA    5,WTO_AR            5 -> WTO BUFFER
*
* RESTORE THE SAVED VALUE TO REGISTER 1
* HERE...
*
         L     3,0(,1)             GET TO PARM LIST POINTER
*
* LOAD THE HALFWORD VALUE AT REGISTER 3 DISPLACEMENT 0 TO REGISTER 4
* HERE...
*
*
* LOAD THE ADDRESS AT REGISTER 3 DISPLACEMENT 2 TO REGISTER 3
* HERE...
*
*
* CHANGE THE WXYZ TO SPECIFY A DISPLACEMENT 0 AND BASE REGISTER 3 IN
* THE MVC INSTRUCTION BELOW.  NOTE THAT FOR THE MVC INSTRUCTION,
* THERE IS NO INDEX PARAMETER (UNLIKE IN LA)
*
LOOP     MVC   OUT_STRING(1),WXYZ
         WTO   TEXT=(5)
         AHI   3,1                 BUMP 3 TO NEXT CHARACTER
         BCT   4,LOOP
LMRET    LM    14,12,12(13)
*
         XR    15,15
         BR    14
* *******************************************************************
* END OF PROGRAM
* *******************************************************************
WTO_AR     DC    H'1'
OUT_STRING DS    C
         LTORG ,
         END
```

Example 3

```
*********************************************************************
* DOT-PRODUCT PROGRAM
* Copyright IBM UK LTD 2013
*********************************************************************
*
* MAIN PROGRAM STARTS HERE
*
EX3      CSECT
EX3      AMODE 31
EX3      RMODE 24
* USUAL PROGRAM SETUP
         STM   14,12,12(13)
         BALR  12,0
         USING *,12
*
* INITIALISE INDEX REGISTER 5 WITH VALUE 0
* HERE...
*
*
* INITIALISE AN ACCUMULATOR REGISTER OF YOUR CHOICE (NOT 12) WITH
* VALUE 0
* HERE...
*
LOOP     DS    0H
*
* LOAD INTO REGISTER 3 THE VALUE OF ARR_R[I] WHERE I IS AN INDEX
* HERE...
*
*
* LOAD INTO ANOTHER REGISTER THE VALUE OF B_ARR[I]
* HERE...
*
*
* MULTIPLY THE REGISTERS TOGETHER
* HERE...
*
*
* ADD THE 32-BIT RESULT TO YOUR ACCUMULATOR
* HERE...
*
         AHI   5,4
         CHI   5,16
*
* BRANCH TO LOOP IF THE CC INDICATES A RESULT OF _LESS_
* HERE...
*
*
* STORE THE RESULT FROM YOUR ACCUMULATOR INTO RESULT
* HERE...
*
LMRET    LM    14,12,12(13)
*
         XR    15,15
         LRL   15,RESULT
         BR    14
* *******************************************************************
* END OF PROGRAM
* *******************************************************************
A_ARR     DC    A(12,3,12,10)
B_ARR     DC    A(4,7,9,8)
RESULT    DC    F'0'
         LTORG ,
         END
```

Example 4

```
*********************************************************************
* SUBROUTINE PROGRAM
* Copyright IBM UK LTD 2013
*********************************************************************
*
* MAIN PROGRAM STARTS HERE
```

```
*
EX1      CSECT
EX1      AMODE 31
EX1      RMODE 24
* USUAL PROGRAM SETUP    <- FIX THIS COMMENT
         STM   14,12,12(13)
         BALR  12,0
         USING *,12
*
* *********************************************************************
* WRITE YOUR CODE HERE
* CALL THE SUBROUTINE MYSUB
* HERE...
* *********************************************************************
*
         LA    5,WTO_AR
         WTO   TEXT=(5)
LMRET    LM    14,12,12(13)
         XR    15,15
         BR    14
* *********************************************************************
* MY SUBROUTINE
* SPECIFICATION:
*     THIS SUBROUTINE SHOULD COPY THE AMOUNT OF BYTES SPECIFIED IN
*     REGISTER 1 AT THE ADDRESS SPECIFIED IN REGISTER 2 TO THE BUFFER
*     SPECIFIED IN REGISTER 3
*     THE ROUTINE SHOULD USE AN MVCL INSTRUCTION IN ORDER TO COPY THE
*     DATA.  INFORMATION ON HOW TO USE THIS CAN BE FOUND IN POPS.
* INPUTS:
*     REGISTER 1  -> LENGTH OF DATA TO BE COPIED
*     REGISTER 2  -> POINTER TO INPUT BUFFER
*     REGISTER 3  -> POINTER TO OUTPUT BUFFER
*     REGISTER 14 -> RETURN ADDRESS
* OUTPUTS:
*     ALL REGISTERS ARE RESTORED
* *********************************************************************
* WRITE YOUR SUBROUTINE CODE HERE
* HERE...
*
* *********************************************************************
* END OF PROGRAM
* *********************************************************************
WTO_AR     DC    H'257'
OUTBUF     DC    257C'0'
BUFLEN     EQU   *-OUTBUF
INBUF      DC    257C'X'
MYSAVEAREA DS    16F
         LTORG ,
         END
```

Answer 1

```
**********************************************************************
* SIMPLE HELLO WORLD PROGRAM
* Copyright IBM UK LTD 2013
**********************************************************************
*
* MAIN PROGRAM STARTS HERE
*
EX1      CSECT
```

```
EX1      AMODE 31
EX1      RMODE 24
* USUAL PROGRAM SETUP    <- FIX THIS COMMENT
         STM   14,12,12(13)
         BALR  12,0
         USING *,12
*
* *********************************************************************
* WRITE YOUR CODE HERE
* MOVE THE DATA IN_STRING TO OUT_STRING
* HERE...
         MVC   OUT_STRING,IN_STRING
* *********************************************************************
*
         LA    5,WTO_AR
         WTO   TEXT=(5)
LMRET    LM    14,12,12(13)
*
* *********************************************************************
* WRITE YOUR CODE HERE
* THE RETURN CODE OF THE PROGRAM IS HANDED BACK IN REGISTER 15
* PROVIDE A RETURN CODE OF 15
* HERE...
         XR    15,15
* *********************************************************************
*
         BR    14
* *********************************************************************
* END OF PROGRAM
* *********************************************************************
IN_STRING  DC    C'HELLO WORLD!'
WTO_AR     DC    AL2(L'OUT_STRING)
OUT_STRING DS    CL(L'IN_STRING)
         LTORG ,
         END
```

Answer 2

```
***********************************************************************
* SIMPLE ADDRESSING LOOP PROGRAM
* Copyright IBM UK LTD 2013
***********************************************************************
*
* MAIN PROGRAM STARTS HERE
*
EX2      CSECT
EX2      AMODE 31
EX2      RMODE 24
* USUAL PROGRAM SETUP
         STM   14,12,12(13)
         BALR  12,0
         USING *,12
*
* SAVE REGISTER 1 SOMEWHERE BECAUSE IT MAY BE USED BY WTO
         LR    3,1
*
         WTO   'HELLO'
         LA    5,WTO_AR           5 -> WTO BUFFER
*
* RESTORE THE SAVED VALUE TO REGISTER 1
```

```
         LR    1,3
*
         L     3,0(,1)            GET TO PARM LIST POINTER
*
* LOAD THE HALFWORD VALUE AT REGISTER 3 DISPLACEMENT 0 TO REGISTER 4
         LH    4,0(,3)
*
*
* LOAD THE ADDRESS AT REGISTER 3 DISPLACEMENT 2 TO REGISTER 3
         LA    3,2(,3)
*
*
* CHANGE THE WXYZ TO SPECIFY A DISPLACEMENT 0 AND BASE REGISTER 3 IN
* THE MVC INSTRUCTION BELOW.  NOTE THAT FOR THE MVC INSTRUCTION,
* THERE IS NO INDEX PARAMETER (UNLIKE IN LA)
*
LOOP     MVC   OUT_STRING(1),0(3)
         WTO   TEXT=(5)
         AHI   3,1                BUMP 3 TO NEXT CHARACTER
         BCT   4,LOOP
LMRET    LM    14,12,12(13)
*
         XR    15,15
         BR    14
* *********************************************************************
* END OF PROGRAM
* *********************************************************************
WTO_AR      DC    H'1'
OUT_STRING DS    C
         LTORG ,
         END
```

Answer 3

```
***********************************************************************
* SIMPLE ADDRESSING LOOP PROGRAM
* Copyright IBM UK LTD 2013
***********************************************************************
*
* MAIN PROGRAM STARTS HERE
*
EX3      CSECT
EX3      AMODE 31
EX3      RMODE 24
* USUAL PROGRAM SETUP
         STM   14,12,12(13)
         BALR  12,0
         USING *,12
         LA    5,0           INITIALISE INDEX REGISTER
         LA    6,0           INITIALISE ACCUMULATOR
LOOP     L     3,A_ARR(5)    LOAD ARRAY A ELEMENT
         L     4,B_ARR(5)    LOAD ARRAY B ELEMENT
         MR    2,4           MULTIPLY RESULT
         AR    6,3           ADD RESULT TO ACCUMULATOR
         AHI   5,4
         CHI   5,16
         BL    LOOP          BRANCH IF NOT AT END OF ARRAY
         ST    6,RESULT      STORE FINAL RESULT
LMRET    LM    14,12,12(13)
*
```

```
         XR    15,15
         LRL   15,RESULT
         BR    14
* ********************************************************************
* END OF PROGRAM
* ********************************************************************
A_ARR     DC    A(12,3,12,10)
B_ARR     DC    A(4,7,9,8)
RESULT    DC    F'0'
         LTORG ,
         END
```

Answer 4

```
**********************************************************************
* SUBROUTINE PROGRAM
* Copyright IBM UK LTD 2013
**********************************************************************
*
* MAIN PROGRAM STARTS HERE
*
EX1      CSECT
EX1      AMODE 31
EX1      RMODE 24
* USUAL PROGRAM SETUP     <- FIX THIS COMMENT
         STM   14,12,12(13)
         BALR  12,0
         USING *,12
*
* ********************************************************************
* WRITE YOUR CODE HERE
* CALL THE SUBROUTINE MYSUB
         LA    1,BUFLEN
         LA    2,INBUF
         LA    3,OUTBUF
         LA    15,MYSUB
         BALR  14,15
* ********************************************************************
*
         LA    5,WTO_AR
         WTO   TEXT=(5)
LMRET    LM    14,12,12(13)
         XR    15,15
         BR    14
* ********************************************************************
* MY SUBROUTINE
* SPECIFICATION:
*     THIS SUBROUTINE SHOULD COPY THE AMOUNT OF BYTES SPECIFIED IN
*     REGISTER 1 AT THE ADDRESS SPECIFIED IN REGISTER 2 TO THE BUFFER
*     SPECIFIED IN REGISTER 3
*     THE ROUTINE SHOULD USE AN MVCL INSTRUCTION IN ORDER TO COPY THE
*     DATA.  INFORMATION ON HOW TO USE THIS CAN BE FOUND IN POPS.
* INPUTS:
*     REGISTER 1  -> LENGTH OF DATA TO BE COPIED
*     REGISTER 2  -> POINTER TO INPUT BUFFER
*     REGISTER 3  -> POINTER TO OUTPUT BUFFER
*     REGISTER 14 -> RETURN ADDRESS
* OUTPUTS:
*     ALL REGISTERS ARE RESTORED
* ********************************************************************
```

```
MYSUB STM  0,15,MYSAVEAREA
      LR   0,2
      LR   5,1
      LA   4,OUTBUF
      MVCL 4,0
      LM   0,15,MYSAVEAREA
      BR   14
* *********************************************************************
* END OF PROGRAM
* *********************************************************************
WTO_AR     DC    H'257'
OUTBUF     DC    257C'0'
BUFLEN     EQU   *-OUTBUF
INBUF      DC    257C'X'
MYSAVEAREA DS    16F
         LTORG ,
         END
```