## Final Exam

<table>
<thead>
<tr>
<th>Multiple Choice</th>
<th>True/False</th>
<th>Extra Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ___</td>
<td>9. ___</td>
<td>1. ___</td>
</tr>
<tr>
<td>2. ___</td>
<td>10. ___</td>
<td>2. ___</td>
</tr>
<tr>
<td>3. ___</td>
<td>11. ___</td>
<td>3. ___</td>
</tr>
<tr>
<td>4. ___</td>
<td>12. ___</td>
<td>4. ___</td>
</tr>
<tr>
<td>5. ___</td>
<td>6. ___</td>
<td>5. ___</td>
</tr>
<tr>
<td>7. ___</td>
<td>7. ___</td>
<td></td>
</tr>
<tr>
<td>8. ___</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-----------------------------

Scoring:

- **Part I** Multiple Choice (24 possible) _______
- **Part II** True or False (8 possible) _______
- **Part III** Short Answers (38 possible) _______
- **Part IV** Coding (30 possible) _______
  - **Total Score** (100 possible) _______
- **Part V** Extra Credit Section 1 (6 possible) _______
  - Section 2 (3 possible) _______
  - Section 3 (21 possible) _______
Part I. Multiple Choice (2 points each).

Choose the best available answer and mark it on the cover page.

1. Which of the following is true about DSECTS?

A. A DSECT is used to set aside an area in storage.
B. A DSECT is used to describe an area in storage.
C. If you code a DC statement as part of a DSECT, object code will be generated.
D. If a DSECT contains executable statements, they will not be assembled, other than to add up the length.
E. More than one of the above is true.

2. When invoking an external subroutine, you must get its address by using a V-Con because:

A. The address of the subroutine is not known during assembly.
B. Because the address is too large for an A-Con.
C. That is the only way to put an address into R15.
D. The BALR instruction expects to have a V-Con in its second argument.
E. None of the above.

3. Suppose we have a 42-byte variable called NOTE, and we want to fill it with copies of C'PQR', so we will have 'PQRPQRPQR...'. What instructions do we need to have to do this?

A. MVI NOTE,CL3'PQR'
   MVC NOTE+3(39),NOTE
B. MVC NOTE(3),=CL3'PQR'
   MVC NOTE+1(41),NOTE
C. MVC NOTE(3),=CL3'PQR'
   MVC NOTE+3(39),NOTE
D. MVC NOTE(42),=14CL3'PQR'
E. At least two of the above should work.

4. Here are several ways to set register 8 to 0. Which of them uses the smallest number of bytes?

A. SR 8,8
B. There is no way to set register 8 to 0.
D. LA 8,0(0,0)
C. L 8,='0'
E. These all use the same number of bytes.
5. The entry linkage at the beginning of a subroutine usually looks like this:

```
STM   14,12,12(13)
LR    12,15
USING NAME,12
LA    14,NAMESAVE
ST    13,4(0,14)
ST    14,8(0,13)
LR    13,14
```

Which of these statements A-C is true?

A. We use 12 instead of 15 as the base register because this subroutine, NAME, might want to call some other subroutine, which change the value of register 15.
B. Register 13 ends up pointing to NAMESAVE.
C. We should not change the values of registers 12 and 13 in NAME's code other than in the linkage.
D. None of these statements A-C is true.
E. All of these statements A-C are true.

6. The exit linkage at the end of a subroutine usually looks like this:

```
L     13,0(0,13)
LM    14,12,12(13)
BR    14
```

If we coded the last line as 'B 14' instead, what would happen?

A. The assembler would catch the error.
B. Nothing would happen: the exit linkage would still work.
C. Execution would continue at absolute address 14, probably not a good idea.
D. Execution would continue at an offset of 14 bytes from the base register.
E. None of the above.

7. PATT is an edit pattern. How can we alter the fill character in PATT to be the colon character C':'?

A. MVI    PATT,=C':'
B. MVC    PATT(1),=C':'
C. ST     PATT(1),=C':'
D. MVI    PATT+1,C':'
E. There is no way to do this—the fill character is always a blank.
8. What information is kept in the rightmost 3 bytes of the PSW?
   A. The condition code, program mask and ILC.
   B. The address of the currently executing instruction.
   C. The address of the save area.
   D. The address of the next instruction.
   E. None of the above.

9. How many length fields does a type SS instruction have?
   A. 0
   B. 1
   C. 2
   D. 3
   E. It depends on the instruction.

10. Under the standard linkage conventions, what can a subprogram always assume upon entry?
    A. Register 13 contains the address of the caller's save area.
    B. Register 15 contains the address for the return to the caller.
    C. Register 14 points to the entry point of the program.
    D. Register 1 contains the backward pointer.
    E. Register 0 is the caller's base register.

11. Which of the following instructions could you use to multiply the number in register 9 by the number 8, without affecting any other register?
    A. MP 9,=F'8'
    B. MR 9,=F'8'
    C. MR 9,8
    D. Any of A -- C could be used.
    E. None -- all of them affect more than one register or have other problems.

12. The correct answer to this question is C.
    A. No, this is not the answer.
    B. This is also not correct.
    C. C stands for correct.
    D. D is decidedly not the one you want.
    E. No, you've gone too far.
Part II. True or False.

Mark your answers on the cover page.

1. All the binary Multiply and Divide instructions require the use of an even-odd pair of registers.
   A. True
   B. False

2. If an instruction has two length fields, each of them has an actual value in the range 1 to 16.
   A. True
   B. False

3. An addressability error can occur for both operands of a CLI instruction.
   A. True
   B. False

4. The instruction MVC 4(4,4),0(6) will copy the 4 bytes located at 4(4) into register 6.
   A. True
   B. False

5. The result of SRP VAL,2,5 on VAL DC PL3'1799' is X'00018C'. (Hint: look at this carefully. Assume the sign is correct.)
   A. True
   B. False

6. An internal subroutine must have its own LTORG.
   A. True
   B. False

7. Before you can use the labels in a DSECT, you must establish addressability to it with a USING statement.
   A. True
   B. False

8. An SS instruction is always 3 half words long.
   A. True
   B. False
Part III. Short Answers (points as marked)

1. (1 point) What is the range of values (from minimum to maximum) for the displacement D we can have in a D(X,B) or D(B) address?

2. (1 point) What is the maximum number of decimal digits can we store in one packed decimal variable?

3. (2 points each) What program interruption code will we have in each case?
   (a) _____ attempt to load a register from an odd-numbered address
   (b) _____ attempt to divide by 0 (binary)
   (c) _____ attempt to multiply with invalid packed decimal data
   (d) _____ attempt to execute an invalid instruction

4. In my program, I have defined the following:

   NUM1    DC F'-14'
   NUM2    DC F'5'
   NUM3    DS F

   Later on, register 4 contains the value F'0' or (X'00000000'). I want to divide NUM1 by NUM2 and put the quotient in NUM3. Here is my code, which contains an error:

   L     5,NUM1
   D     4,NUM2
   ST    5,NUM3

   (a) (4 points) What values (all 8 hex digits) are in registers 4 and 5 after the code is executed? (Be careful. Hint: hex 'F' is decimal 15, and 15 divided by 5 is 3.)

   Reg. 4: __ __ __ __ __ __ __ __  Reg. 5: __ __ __ __ __ __ __ __

   (b) (2 points) How can I avoid this mistake?
5. (2 points each) For each of the following DC instructions, write all of the contents of the field in hexadecimal, one hex digit per blank. Do not write anything more than that. If you need a sign digit, use any appropriate value.

VAL1  DC  PL4'-198346' __ __ __ __ __ __ __

VAL2  DC  F'-8' __ __ __ __ __ __ __

VAL3  DC  ZL4'107' __ __ __ __ __ __ __

VAL4  DC  2CL3'MH' __ __ __ __ __ __ __ __ __ __ __

VAL5  DC  B'10010011' __ __

6. (4 points) Here is code someone has written at the beginning of a subroutine called UNDER, which is receiving 3 parameters A, B and C. UNSAVE is an 18-FW save area defined later in UNDER. This code contains several errors. The answer to question 9 is B. Find 2 errors (which line) and what is wrong (give a correction). (You only need to find 2 errors. There are plenty.)

UNDER  CSECT                     Line 1
STM   14,12,12(14)           Line 2
LR    15,12                  Line 3
USING 12,UNDER              Line 4
LA    13,UNSAVE              Line 5
ST    13,8(14)               Line 6
ST    14,4(13)               Line 7
LR    14,13                  Line 8
STM   4,6,0(1)               Line 9

Error 1

Line: ________________________________
Correction: __________________________

Error 2

Line: ________________________________
Correction: __________________________
7. Suppose program MAIN is calling a subprogram called SUB, and SUB is calling a subprogram called CALC. Each of these has a save area: MAINSAVE, SUBSAVE, and CALCSAVE, respectively. SUB and CALC have parameter lists. We are using standard linkage.

(Hint: The code in question 6 is not a good guide to answering this.)

(a) (2 points) After the entry linkage of CALC, exactly what piece of information is stored at CALSAVE+4?

(b) (2 points) After the entry linkage of SUB, exactly what piece of information is stored at MAINSAVE+8?

(c) (4 points) At the very beginning of CALC, before the entry linkage, exactly what is in each of these registers?

Register 1: ___________________________________________

Register 13: ____________________________________________

Register 14: ___________________________________________

Register 15: ___________________________________________
Part IV. Coding (points as marked).

1. (4 points) A subroutine called JUNIOR should return a calculated value in register 0, and we also want it to provide a return code in register 15. The values we want have already been put in registers 0 and 15. The exit linkage for JUNIOR starts with

   L 13,4(13)

and ends with

   BR 14

Write the code that goes between them.

2. (6 points) I have a 9-byte zoned decimal field at address 10(12). I want to convert this zoned decimal value to a binary value in register 9.

Write the code needed to do this. You may need to define a variable.
3. (5 points) Suppose we have the following defined in a program:

```
PNUM    DS    PL3
OUTNUM  DS    CL7
```

At some point in our program, we want to edit PNUM and put the result in OUTNUM. Assume PNUM contains an integer (no decimal point). We want to suppress leading zeroes. If PNUM is negative, we want to print a minus sign (-) immediately to the right of the last digit. If PNUM is zero, we do want to print a zero.

Write the code we need to do this.

4. (8 points) Suppose we have declared the following:

```
BIGLIST   DS   70CL32      a table of 70 entries, 32 bytes each
```

At some point in our program, register 5 contains the index of a table entry, a positive number (1 to 50). We want register 7 to contain the address of a corresponding entry in the table.

That is, the first entry is at index 1 and address BIGLIST. The sixth entry is at index 6 and address BIGLIST+160.

That is, we want to convert an index into an address.

Write the code we need to do this. You may change the value of registers 4 and 5 if necessary. Do not change the value of registers other than 4, 5 and 7. Note: 32 = 2 to the 5th power.
5. Suppose a table has been defined:

```
CUSTABLE DC 0F
DS 120CL20     a table of 120 entries, 20 bytes each
```

Each entry will contain these pieces of information, in this order (and nothing else):

- Customer ID -- 10 characters
- Customer FFM -- 4 bytes, binary (FFM = Frequent Flier Miles)
- Customer Balance -- 6 bytes, signed packed decimal (dollars and cents)

(a) (4 points) Define a DSECT that could be used to describe one table entry. (Invent reasonable names.)

(b) (2 points) Now give the assembler code that is needed to put the address of CUSTABLE in register 7 and establish addressability to your DSECT so that it describes the first entry of CUSTABLE.

(c) (1 point) Much later in the program, we will be done with the DSECT. Give the code we need to detach it.
Part V. Extra Credit.

Section V-1. Multiple Choice (1 point each).

Choose the best available answer and mark it on the cover page.

1. Suppose I have the following in storage:

TREAT       DC   256C'
ORG   TABLE+C'a'
DC      CL5'ABCD'E'
ORG

Suppose I have just read an 80-byte record called INBUFF using XREAD. The record contains someone's answers to a multiple-choice test, so it should contain only spaces and upper-case letters 'A' through 'E'. Unfortunately, a user may have supplied lower case letters 'a' through 'e'. Nothing else is wrong with the data. The line might look like this:

A B c D e a D D c C b A E d C

I execute this instruction:

TR   INBUFF(80),TREAT

What is the result?

A. Lower-case letters 'a' through 'e' in INBUFF are changed to upper-case.
B. Anything in INBUFF except 'a' through 'e' is changed to a blank.
C. The table will not assemble, so nothing will happen.
D. This is a disaster, as I should be using TRT instead.
E. More than one of the above is true.

2. We have a 1-byte variable called BYTE. How can we change the first and last bits of BYTE to 0 but leave the rest of BYTE alone?

A. OI   BYTE,B'01111110'
B. NI   BYTE,B'01111110'
C. XI   BYTE,B'01111110'
D. NC   BYTE(1),=X'7E'
E. Two of the above will work.
3. The prototype statement of a macro:
   A. Is the second line of the macro definition.
   B. Determines the format you must use to invoke the macro.
   C. Can contain positional and/or keyword parameters.
   D. May start with an optional label (&LABEL).
   E. All of the above.

4. The TR instruction:
   A. Is used to search for certain hexadecimal values in a string of characters.
   B. May alter the contents of registers 1 and 2 when it executes.
   C. Will stop scanning when it finds a character whose entry in the table is X'00'.
   D. Requires the use of a 256-byte table.
   E. More than one of the above.

5. Which of the following is true concerning the EX instruction?
   A. It can only be used with the instructions TRT, MVC, TM.
   B. It is used to clear an area of storage to binary Os.
   C. It is the only way to move asterisks.
   D. It provides an easy way to pack or copy a variable length field.
   E. It can make permanent changes in the source code.

6. Suppose we have a bit called M. Which of the following is true?
   A. M AND 1 gives the result 1.
   B. M OR 1 gives the result 0.
   C. M XOR 0 gives the result 0.
   D. M XOR M gives the result 0.
   E. M OR M gives the result 1.
Section V-2. True or False (1 point each).

Mark your answers on the cover page.

1. If I use TRT to search a character variable for just the letter C'Z', I need a search table in which most entries have the value X'00'.
   
   A. True
   B. False

2. We could always use STCM instead of ST to copy a register’s value into a fullword of memory.
   
   A. True
   B. False

3. We can multiply the signed 64-bit integer in registers 5 and 6 by $2^{17}$ (that is, 2 to the 17th power) by using the instruction SLDA 5,17. (Look at this carefully.)
   
   A. True
   B. False

Section V-3. Short Answers and Coding (points as marked).

1. Suppose I have the following in my program:

   WHATISIT DR 0,0

   Now suppose register 7 contains the value F'147' = X'93', and I execute the following instruction:

   EX 7,WHATISIT

   (a) (2 points) Write out the object code for the instruction at WHATISIT in hex digits.

   (b) (2 points) When EX is executed, a copy is made of the instruction at WHATISIT, with a slight change. Write out the new copy in hex digits.
1 continued.

(c) (1 point) Now decode the instruction you just wrote out in (a).

(d) (1 point) When an attempt is made to execute the new instruction, what will happen?

2. (2 points each) For each of the following, write one instruction to do what is indicated. You may need a literal.

(a) Store the two middle bytes of register 11 at the label MID.

(b) Shift all the bits of register 11 to the right by 8 positions.

(c) Multiply the number in register 8 by 2 to the 9th power (which is 512). Do not change the value of any other register. Assume the product will fit into register 8.

(d) We have a 2-byte variable called TWYTE. Change the first 3 bits and last 3 bits of TWYTE to 1 but leave the rest of TWYTE alone. (Notice TWYTE is two bytes long.)

(e) Test the fifth bit (counting from the left) of the byte at address 15(7) to see whether it is a 1 or a 0.
3. (5 points) We want to have a macro in our program that will set a fullword value = the lesser of two other fullword values. Here is the code for it, but some parts have been left blank. Fill in the blanks. Do not worry about overflow.

MACRO

&LABEL MIN &FIRST,&SECOND,&RESULT
AIF ('&FIRST' EQ '').NOPARM
AIF (_________ EQ '').NOPARM
AIF ('&RESULT' EQ '').NOPARM

&LABEL B _________

L&SYSNDX DS F
M&SYSNDX ST 4,L&SYSNDX
L 4,&FIRST
____ 4,&SECOND
BNH P&SYSNDX
L 4,_________
P&SYSNDX ST 4,&RESULT
L 4,L&SYSNDX
AGO .END

.NOPARM ANOP

MNOM 'A parameter is missing.'

.END ANOP

___