Afifah Arif

CSCI 360

This assignment is practice in reading a dump.

Type your answers to these questions and submit them on Blackboard by the due date.

Be sure to include your name.

Show your work!

Run the following program. You can use the JCL we used in the lab training exercise.

```
DEBUG1   CSECT
         USING DEBUG1,15    Establish a base register
         L     3,FIRST      Load a number into register 2
         L     4,SECOND     Load another number into register 4
         SR    3,4          Add the two numbers
         LA    5,THIRD      Register 5 will = address of THIRD
         L     6,CONST      Put CONST's value in register 6
         ST    3,0(5,6)     Store the result
         XDUMP FOURTH,4     Dump the result
         SR    6,6          Set register 6 to 0
         BCR   B'1111',14   Return to caller
* This is a comment line.
NOTE     DC    CL19'STORAGE STARTS HERE'
CONST    DC    F'2'         A constant = 2
FIRST    DC    F'193'       A number
SECOND   DC    F'89'        Another number
THIRD    DS    XL4'00'      Yet another number
FOURTH   DS    F            Difference of FIRST and SECOND
         END   DEBUG1       Program ends here
```

Use the resulting output to answer the following questions:

1. (1 point) Did this error occur (a) while the program was being assembled or (b) when it was being run?

   The error occurred while the program was running.

2. (2 points) What is the address of the next instruction which will be executed?

   The address of the next instruction is 000016.

3. (2 points) What is the value of the condition code at the time of the ABEND?

   The value of the condition code is 00.

4. (2 points) What is the length of the instruction that caused the ABEND (a number of bytes)?

   The length of instruction that caused the ABEND is 4 bytes long.

5. (2 points) What is the address of the instruction that caused the abend?

   The address of the instruction that caused the ABEND is 000016-2 = 000014.

6. (2 points) What type of error occurred (number and name)?

   The type of error that occurred was a specification exception with the number 0006.

7. (2 points) What actually causes this error?

   The cause of this error is that the calculated address is not on a full-word boundary. In other words, it is not divisible by 4.

8. (3 points) Correct the error by rewriting the section of code that caused it. (The difference should be stored in FOURTH.) (There are several correct ways to do this.)

```
L     3,FIRST      Load a number into register 2 => L     2,FIRST

SR    3,4          Add the two numbers => AR    4,2

ST    3,0(5,6)     Store the result => ST    4,0(5,6)
```

9. (2 points) At the time of the ABEND, what is the value of register 3 in decimal?

   The value of register 3 would be restored to its original value at the time of the ABEND which is a decimal value of 00.

10. (2 points) What does the value in register 3 represent at the time of the ABEND?

    The value in register 3 represents the original value of THIRD at the time of the ABEND.

11. (3 points) Why is the address (LOC column) of the storage area with the label CONST on it at X'000034' when the DC statement whose address is X'000020' only takes up 19 bytes?

    The address of CONST is at X'000034' because NOTE has character data which is not aligned on any boundary.

12. (2 points) What are the contents of the byte of user storage starting at address X'000033'? What do they represent?

    The contents of the full-word beginning at X'000033' are ‘5D9C5F5’and represent the beginning of the storage area.

13. (1 point) Why do we not have XDUMP output?

    There was no output because the file was unopened, and the cards were not read.

14. (2 points) How many bytes are taken up by the XDUMP pseudo-instruction?

    The XDUMP instruction takes up 6 bytes.

15. (2 points) If we correct the error, what should be the value of the condition code at the end of the program (when we reach the BCR line)?

    The condition code should be 1.