Assignment 2

80 points

For this assignment you will write a program that reads in an unknown number of lines, each containing 3-digit numbers, 4 per line. For each line read, print out the numbers W, X, Y and Z, calculate the value of W - X + Y - Z and print the result, single-spaced.

Each line has the following format:

```
columns 1-3    blank
columns 4-6    first number W
columns 7--9   blank
columns 10-12  second number X
columns 13-15  blank
columns 16-18  third number Y
columns 19-21  blank
columns 22-24  fourth number Z
columns 25-80  blank
```

When all processing is done, skip a line and then print the sum of the result values from all of the lines and the number of lines read.

Write your program incrementally! That means that you should begin by just reading one record and print out those numbers. When that works, put in a loop. If you get one part working before moving on to the next, your debugging will be much easier and less time-consuming.

Use a top driven loop:

```
Initialize the counter and the total
     Top of the loop
      Read a record
      If end-of-file, branch to the end of the loop
      Deal with the record just read using XDECI, arithmetic, XDECO and XPRNT
      Add 1 to the counter
      Add the result (W - X + Y - Z) of the numbers read to the total
      Branch to the top of the loop
     End of the loop
     Use XDECO and XPRNT to print the summary lines
```

You will need to put labels on two lines, one for the top of the loop and one for the bottom of the loop. You can actually put a label on any line of code, but many people put them on lines that don't do anything else, like this:

```
MYLABEL  DS    0H
```

Here DS 0H takes up no space. (It declares 0 halfwords on a halfword boundary, and as each instruction is an even number of bytes, the location will already be on a halfword boundary.)

JCL for this assignment

Use the following JCL:

```
//KCnumberA JOB ,'Your Name',MSGCLASS=H
//STEP1    EXEC  PGM=ASSIST
//STEPLIB    DD  DSN=KC02293.ASSIST.LOADLIB,DISP=SHR
//SYSPRINT   DD  SYSOUT=*
//SYSIN      DD  *
```

```
************************************************************
*
*  Program:     ASSIGN2
*  Programmer:  Your Name
*
*  Register usage:
*
************************************************************
(Your program goes here.)
/*
//FT05F001   DD  *
   005   005   005   005
   006   001   002   004
     0    +7     2   -45
   100   088   035    10
     0     0     0     0
    10   -10    10   -10
   500   230     9    58
   516   853     0    17
   020     0   245   316
   529   977   681     0
   013   250    85   831
     0   364   275     0
   887   100   293   993
   234   447   591    13
   -89    -7    23   104
   001   002   003   004
   008   007   006   005
   999   998   997   996
/*
//FT06F001   DD SYSOUT=*
//
```

As before, you will need to replace "Your Name" with your own name, and you will need to replace "KCnumber" with your own logon ID.

Your actual code should be after the comment box and before the /* line.

Data

As indicated above, the data is listed in the source code file between the FT05FT001 line and the following /* line.

This is known as instream data. It is also possible to read from a specific disk file by name, and we will do so later.

Other requirements

In the JCL, at the very beginning of the program is a comment box. Notice the place that says "Register Usage". Make a list here of registers you used and how you made use of each one. For instance, you will be using register 15 as your base register, and register 1 is used by XDECI. Thus you might have:

```
*
*  Register usage:
*    1        Used by XDECI
*   15        Base register
*
```

and probably several more such lines.

The comment box should also list your name and the number of the assignment (Assignment 2).

To use XREAD, you will need to have an 80-byte field to contain each line you read.

To produce the output, you will need to define a couple of output lines containing DC and DS statements (with labels on the DS statements).

Your program should include line documentation. At the end of each line (certainly for most lines) skip one or more spaces and insert a few words describing what that instruction does. Try to line these up so they start in the same column. For instance:

```
        SR    2,2                   Initialize counter.
        SR    6,6                   Initialize total.
```

Name your program file something like "ASSIGN2" or "ASSIGN2".

Submit your program file and output file through Blackboard.