Assignment 5

100 points

Overview

In this assignment, you will rewrite the program you wrote in Assignment 4 using external subroutines instead of internal subroutines. You will need to use standard entry linkage and exit linkage in the main routine and in each subroutine.

There are a few minor changes and one additional subroutine.

Input

The input to the program will be a file with an unknown number of records.

The data starts with a number of lines very much like those in the data for Assignment 4, followed by a line containing the integer -12345678. This number serves as a delimiter and is not part of the data. After that are several lines each containing one small nonzero integer.

Use the following JCL statement to specify the input file:

```
//FT05F001  DD  DSN=KC02314.SPRING19.CSCI360.HW5DATA,DISP=SHR
```

What needs to be done

- Call the BUILD subroutine to read lines in the file until it find the delimiter line. Each line contains zero or more integers. Put the integers in the table. Do not put the delimiter value -12345678 in the table. Return to the main program.
- After that, in the main routine, print a heading saying something like "Table Contents".
- Now call subroutine PRINT to print the contents of the table with 7 numbers on each line.
- Next, call the TALLY subroutine which will read the rest of the file.

External Subroutines

You will need several external subroutines:

- BUILD is a subroutine that will read the input file and build the table. It will store the address of the last entry in a fullword passed in as its second parameter.

  The arguments for BUILD are:

  - the address of the input buffer
  - the address of the table
  - the address of a fullword containing the address of the first available entry

- PRINT is a subroutine that will print the numbers in the table with 7 values per line, starting on a new page and double-spaced.

The arguments for PRINT are:

  - the address of the table
  - the address of a fullword containing the address of the next available entry

- TALLY is a subroutine that will count numbers in the table that are multiples of the last few numbers in the file.

  The arguments for TALLY are:

  - the address of the input buffer
  - the address of the table
  - the address of a fullword containing the address of the first available entry

There are several requirements for using an external subroutine:

- Each external subroutine is actually a separate program. You need a CSECT with the name of the subroutine, as in:

  ```
  BUILD    CSECT
  ```

- You need to create a parameter list for the subroutine (a set of consecutive fullwords, each containing the address of a parameter). We did this with internal subroutines.

- You need to call the subroutine, as in:

  ```
  LA      1,BPARMS         Parmlist for BUILD
  L       15,=V(BUILD)     Get the address of BUILD
  BALR    14,15            Branch to BUILD
  ```

  Here the BALR instruction will set register 14 = the address of the next instruction after the BALR.

- In the subroutine, you need a register save area and standard entry linkage. (You should probably go read about standard entry linkage.)

- At the end of the subroutine, you need standard exit linkage. (You should probably go read about standard exit linkage.)

What does the TALLY subroutine do?

The TALLY subroutine will read each line in the file after the delimiter. For each line, it should do the following:

- extract the integer M on the line using XDECI
- in a loop, go through the table and count the multiples of M
- print a line stating the number of multiples of M in the table

---

Other Requirements

- The JCL for this assignment is the same as the JCL for Assignment 4 except the line above to specify the input file.

- You may assume that the table needs to hold no more than 80 values. You do not need to count the numbers. Each entry is one fullword. Initialize the table to the value -32 (as in 80F'-32').

- In PRINT, double-space between lines of numbers.

- In TALLY, triple-space the lines of output.
- You may use register equates if you want, but it is not required.
- You may use extended mnemonics such as BH, BL, BNE, etc. for branch instructions.
- Document your program as usual and submit it through Blackboard as usual.