LAB #9 [10 pts]

Bitwise operations for Big Data

Overview

Let's pretend you are working for a company that does big data analysis based on large human subject datasets (e.g., Turkish Census data, surveys). From such datasets, boolean features of each person have been extracted that are relevant for a particular data analysis task.

Each data item in the dataset is a record with 4 booleans:

- **b1:** true if the person is under 40, false otherwise.
- **b2:** true if the person has an Android phone, false if they have an iPhone.
- **b3:** true if the person prefers Netflix, false if they prefer Disney; and
- **b4:** true if the person prefers Star Wars, false if they prefer Star Trek

You will implement a program, adding features incrementally, that processes 32 bits. These 32 bits are provided as keyboard input as an integer. Each digit (4 bits) represents a single person whether he is old or young; he has Android or iPhone; he uses Netflix or Disney as streaming service; he likes Star Wars or Star Trek. In a real-world scenario, one would of course get the input from a (huge) binary data file.

Implementation constraints

- **Constraint #1:** Your .data segment must be **exactly** as shown below, otherwise you will get no credit for this assignment:

```
segment .data
    msg1    db      "Enter an integer: ", 0
    msg2    db      "Binary representation: ", 0
    msg3    db      "Semantic: ", 0
    msg4    db      "YNGR/IPHN/HULU/SWRS count: ", 0
    msg5    db      "*/ANDR/*/STRK count: ", 0
    codes   db      "YNGR", 0, "OLDR", 0, "ANDR", 0, "IPHN", 0, "NFLX", 0, "DISN",
0, "SWRS", 0, "STRK", 0
    four_spaces db  "    ",0
```

- **Constrained #2:** Your .bss segment must not reserve more than 8 bytes, otherwise you will be no credit for this assignment.

```
segment .bss
                input    resd  1     ; only 4 bytes!
```

Printing semantics of input integer

Augment your program so that it also prints the *semantics* of the input integer. Each integer has information for 8 people. The program should print each of the four boolean values as a 4-character string using the following mapping:

boolean	**0**	**1**
b1	YNGR	OLDR
b2	ANDR	IPHN
b3	NFLX	DISN
b4	SWRS	STRK

The program should print the "decoding" on a line, starting with 4 white spaces. Here are example interactions with the program, which you should match:

```
Enter an integer: 512331
Binary representation: 0000 0000 0000 0111 1101 0001 0100 1011
Semantic:
    YNGR ANDR NFLX SWRS
    YNGR ANDR NFLX SWRS
    YNGR ANDR NFLX SWRS
    YNGR IPHN DISN STRK
    OLDR IPHN NFLX STRK
    YNGR ANDR NFLX STRK
    YNGR IPHN NFLX SWRS
    OLDR ANDR DISN STRK
```

```
Enter an integer: 257571913
Binary representation: 0000 1111 0101 1010 0011 1100 0100 1001
Semantic:
    YNGR ANDR NFLX SWRS
    OLDR IPHN DISN STRK
    YNGR IPHN NFLX STRK
    OLDR ANDR DISN SWRS
    YNGR ANDR DISN STRK
    OLDR IPHN NFLX SWRS
    YNGR IPHN NFLX SWRS
    OLDR ANDR NFLX STRK

Enter an integer: -1
Binary representation: 1111 1111 1111 1111 1111 1111 1111 1111
Semantic:
    OLDR IPHN DISN STRK
    OLDR IPHN DISN STRK
    OLDR IPHN DISN STRK
    OLDR IPHN DISN STRK
    OLDR IPHN DISN STRK
    OLDR IPHN DISN STRK
    OLDR IPHN DISN STRK
    OLDR IPHN DISN STRK
```