CPS-360, Assignment #5, 10 points

Due:

Goal: Exploration of 1-bit error detection and correction (ECC) per Hammings's code.

Statement: Do all work in directory ~/360/a4. Write structured, modular and properly documented C-program. Place the program in file a3.c. Also write a Makefile to generate the executable (modify the one used in previous assignment).

Input to be obtained from stdin (scanf() is your friend).

The program will be invoked as:

```
ecc < inputfile
```

where inputfile contains one word (integer) per line of input. The program will read one word at a time (till EOF condition) and process it:

- . extract lower 16-bits message (to encode)
- . encode the message using Hammings code under even parity scheme (determine the redundent bits - the result will be 21-bit coded message)
- . and print the result in the specified format (see sample output).

Sample output:

```
uncoded-message:  1101101010111101
redundent bites:  1:0 2:0 4:0 8:0 16:0
coded message:    001010101010101011101

uncoded-message:  1000100010001000
redundent bites:  1:1 2:0 4:1 8:0 16:1
coded message:    101100001000100101000
etc.
```

Modules: parameters as needed, no globals allowed

- main(?): of course
- xtractmessage(?): extracts the lower order 16-bits into an array of char
- makemessage(?): using even parity scheme generate the redundant bits
- printrsult(?): write the result to stdout.

Turn-in: e-mail attached file named <globalid>-a5.tar.bz2. Do follow the guidelines.