CS 220, Fall 2017
Assignment 6 – (100 points)
Due date: 15 Oct 2017, 11:59pm.

Instructions

- Answer the questions individually. Group effort is not allowed.
- Solutions must be committed to your respective repositories on github.
- For this assignment, you are not allowed to use any library functions other than printf.
- Ensure that your code runs on remote.cs.binghamton.edu.
- Prototypes must be provided for all functions within the header file `define.h`.
- Code must be appropriately commented.
- Useful resources:
  - Common linux commands: `http://www.informit.com/blogs/blog.aspx?uk=The-10-Most-Important-Linux-Commands`
  - `http://c-faq.com/`
  - `https://cdecl.org/`

Questions

1. **(100 points)** Programs operate on files with specific format. For example, Linux executes ELF binaries. Microsoft Windows executes PE binaries. ELF is a file format that tells the loader where and what to find inside the file. Similarly, a word processor can open .doc files, Zip application can open .zip, .gz, etc. In order to aid in detecting if a file is of a particular type, a signature is embedded within the file to indicate the type of a file. Often, the signature is the first few bytes within a file. This will enable an application to quickly identify if it can operate on the file. In this problem, you will implement a program `file_recognizer` that accepts a file as command line input, and prints the

Table 1: Signature to type mapping. All signatures start from the beginning of the file. This is only a sample for the assignment. For curious minds, more can be found here: https://en.wikipedia.org/wiki/List_of_file_signatures

Signature (first few bytes in Hex)	**File type**	**Expected output**
47 49 46 38 39 61 or 47 49 46 38 37 61	GIF Image	This is a GIF file.
7f 45 4C 46	ELF file	This is an ELF file.
25 50 44 46	PDF file	This is a PDF file.
50 4B	ZIP file	This is a ZIP file.
CA FE BA BE	Java class file	This is a Java class file.
89 50 4E 47 0D 0A 1A 0A	PNG file	This is a PNG file.
Anything else	Unknown file	File type unknown.

format of the file and the program that can open it. NOTE: Although file extensions can sometimes tell what format a file is, it is an unreliable indicator. The file signature is the true indicator of a file type.

2. You will implement `file_recognizer.c` and `file_recognizer.h` that contains all your implementation. You will write a Makefile that generates `file_recognizer` executable. Your program must accept exactly one command line argument (i.e., file name). If more or less than expected number of arguments are passed, you are to print ”Invalid input.” and exit.

Useful resources

1. Examine `fopen()`, `fscanf()` and `fclose()` functions to perform file IO.

2. Run the `find` command (e.g., `find / -name "*.png"`) on remote.cs.binghamton.edu to find files of specific types to test.

Testing your code

In order to test your code, we will run the make command to generate the program. We will then call the program in a loop on a set of known file types, and in each case, we will check if the output is as expected in Table 1. We will also provide invalid number of arguments and non-existent files as arguments. NOTE: The testing process may be automated. It is your responsibility to ensure that the files are named correctly. You are advised to test the programs before submitting. Each valid input file is guaranteed to be at least 10 bytes in size.