1: GOAL

To achieve the same functionality as the unix cat command. That is, being able to echo any number and size of input files to the standard output, or to redirect it to another file. Usage would be to quickly view files, concatenate files, and create files.

2. ASSUMPTIONS

Cat can print anything, including binary files and pdfs. Output may not be readable.

Cat is not able to print directories.

3. DESIGN

Use UNIX system calls to open files as file descriptors, read and write files. Usage of cstrings and arrays to handle input arguments

ex. int main(int argc, char** argv)

Functionality to print to standard output is encapsulated into a function named echo().

Error checking is done by returning an integer, that corresponds to an error code, and then printing the error to standard output.

4. PSEUDOCODE

```
func echo(file descriptor):
    do:
        system read to bytesRead from file descriptor
        error check on bytesRead //return 1, 2, 3...
        system write bytesRead to standard out
        error check on write //return 4, 5. 6...
    while: bytesRead > 0 // stop if bytesread is 0 aka EOF is reached
    success: return 0

func main(argc, argv): //argc is # of args, argv is argument values
    if argc = 0:
        echo(stdin)
    else:
        for arg in argv:
            system open argv as filedescriptor
            error checking on open=
            echo(file descriptor)
            system close argv
    return
```