1.
a) Minimum Match: At least one character a-z, then any single character, then 0
example: aA0

b) Minimum Match: Only any single character
example: a

c) Minimum Match: Any single digit 0-9
example: 0

d) Minimum Match: Anything matches
example: (That's a blank as in anything matches)

2.
a) ^(0b)?[01]+$

b) 0

3.
a) ^[+-]?[0-9]*[02468]$

b) 0

4.
a) ^([+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+))|([+-]?[0-9]+([eE][+-]?[0-9]+))?$

b) .1

5.
a) wget http://courses.eece.maine.edu/ece331/reccar.tar.gz

b) tar -xzvf reccar.tar.gz

6.

```
#!/usr/bin/python
import os

# Gets file paths sorted
path = 'reccar/'
files = sorted(os.listdir(path))

# Goes through each file and find the MPH for the winds that day
# On Mount Katahdin
for name in files:
    try:
        f = open(os.path.join(path, name), 'r')

        # Prints file name but without UTC.txt
        file = name[:name.find(' UTC')]
        print file,

        # Reads data and get only Katahdin data
        winds = ""
        data = f.read()
        katahdin = data[data.find('MOUNT KATAHDIN'):data.find('$$')].upper().replace("\n","
 ")

        # Finds the MPH and locate the wind speed
        if katahdin.find('MPH') != -1:
            winds = katahdin[katahdin.find('MPH') - 10:katahdin.find('MPH')]
            if winds.find('TO') == -1:
                winds = winds[winds.find('MPH') - 4:]
            if winds[:1].isalpha():
                winds = winds[1:]

        # Gets rid of any extra spaces due to single digits and newlines
        # Also gets rid of TO if it has it
```

```
        if winds[:1] == " ":
            winds = winds[1:]
        if winds[:] == "":
            print "0 0"
        else:
            wind = winds.replace("  "," ")
            print wind.replace(" TO "," ")

        f.close()
    except:
        print "Cannot open",name
```

7.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <sys/types.h>
#include <sys/stat.h>

void fileinfo(char file[]);

int main(int argc, char *argv[])
{
        // Checks for correct usage
        if(argc == 2) {
                fileinfo(argv[1]);
        }
        else {
                printf("Usage: ./fileinfo (Filename)\n");
        }
}

void fileinfo(char file[])
{
        // stat struct
        struct stat sfile;

        // Checks for any stat error
        if (stat(file, &sfile) == -1) {
                printf("Error reading file info!\n");
        }

        // Gets file type and prints
        printf("\nFile Type\n");
        printf(S_ISBLK(sfile.st_mode)?"Block Special File":"");
        printf(S_ISCHR(sfile.st_mode)?"Character Special File":"");
        printf(S_ISDIR(sfile.st_mode)?"Directory":"");
        printf(S_ISFIFO(sfile.st_mode)?"Pipe or FIFO Special File":"");
        printf(S_ISREG(sfile.st_mode)?"Regular File":"");
        printf(S_ISLNK(sfile.st_mode)?"Symbolic Link":"");
        printf("\n");

        // Gets user, group, and other permissions and prints
        printf("\nFile Permissions User\n");
        printf((sfile.st_mode & S_IRUSR)? "r":"-");
        printf((sfile.st_mode & S_IWUSR)? "w":"-");
        printf((sfile.st_mode & S_IXUSR)? "x":"-");
        printf("\n");
        printf("\nFile Permissions Group\n");
        printf((sfile.st_mode & S_IRGRP)? "r":"-");
        printf((sfile.st_mode & S_IWGRP)? "w":"-");
        printf((sfile.st_mode & S_IXGRP)? "x":"-");
        printf("\n");
        printf("\nFile Permissions Other\n");
        printf((sfile.st_mode & S_IROTH)? "r":"-");
        printf((sfile.st_mode & S_IWOTH)? "w":"-");
        printf((sfile.st_mode & S_IXOTH)? "x":"-");
```

```
        printf("\n");

        // Prints all permissions including file type
        printf("\nPermissions\n");
        printf(S_ISBLK(sfile.st_mode)?"b":"");
        printf(S_ISCHR(sfile.st_mode)?"c":"");
        printf(S_ISDIR(sfile.st_mode)?"d":"");
        printf(S_ISFIFO(sfile.st_mode)?"p":"");
        printf(S_ISREG(sfile.st_mode)?"-":"");
        printf(S_ISLNK(sfile.st_mode)?"l":"");
        printf((sfile.st_mode & S_IRUSR)? "r":"-");
        printf((sfile.st_mode & S_IWUSR)? "w":"-");
        printf((sfile.st_mode & S_IXUSR)? "x":"-");
        printf((sfile.st_mode & S_IRGRP)? "r":"-");
        printf((sfile.st_mode & S_IWGRP)? "w":"-");
        printf((sfile.st_mode & S_IXGRP)? "x":"-");
        printf((sfile.st_mode & S_IROTH)? "r":"-");
        printf((sfile.st_mode & S_IWOTH)? "w":"-");
        printf((sfile.st_mode & S_IXOTH)? "x":"-");
        printf("\n");

        return;
}


# Makefile for fileinfo.c

TARGET=fileinfo
OBJS=fileinfo.o
CFLAGS=-Wall -g
SOURCES=fileinfo.c

all: $(TARGET)
$(TARGET):
        gcc -o $(TARGET) $(CFLAGS) $(SOURCES)

clean:
        rm -f $(TARGET) $(OBJS) core*
```

8. enscript --header='$n %E %*|$%|Spencer Goulette' hw03.txt -o - | ps2pdf - output.pdf