```
#!/usr/local/bin/perl
#THIS FILE WAS GENERATED FROM lang.src -- DO NOT EDIT!

# lang: utility program used by prettyp to detect the language for a
# given file. It uses the filename, the extension of a given file, or
# failing that the magic (#!) line for the interpreter, and outputs
# the language definition in the vgrindefs database. Exit code of 0 if
# detection was sucessful else 1.

#Usage: lang filename

# by Luis Fernandes <elf@ee.ryerson.ca>
# $Id: lang.src,v 1.11 2000/02/16 17:10:21 elf Exp $

# alist that maps filename extension/interpreter => language
# "filename extension", "language in vgrindefs"

%lang = (
                "mod", "model",        "m", "model",
                "p", "pascal",
                "c", "C",                       "h", "C",
                "C", "C++",            "cpp", "C++",    "cxx", "C++",
                "cc", "C++",           "hh", "C++",
                "java", "java",
                "isp", "ISP",           "i", "ISP",
                "sh", "sh",             "csh", "csh",   "zsh", "sh",
                "ksh", "sh",            "bash", "sh",   "tcsh", "csh",   # interpreter
                "ldl", "LDL",
                "I", "ICON",            "icn", "ICON",
                "el", "MLisp",          "ml", "MLisp",
                "rat", "ratfor",        "r", "ratfor",
                "m2", "modula2",        "y", "yacc",
                "f", "fortran",         "py", "python",
                "p", "perl",            "pl", "perl",
                "py", "python",        "python", "python",
                "perl", "perl",         "wish", "tk",                   # interpreter
                "wish -f", "tk",                                                # alternate interpr
                "tcl", "tcl",           "tk", "tk",
                );

#validate number of args
if($#ARGV < 0 ){
        printf STDERR "Usage: lang filename\n";
        exit 1;
}

#Multiple files can be given on the command line (FOR TESTING ONLY!)
foreach(@ARGV) {

        if(! -e) {
                printf STDERR "lang: file $_ does not exist.\n";
                exit 1;
        }
```

```
        #First check for filename matches
        if(/[Mm]akefile/){
#               printf STDERR "lang: $_ is Makefile\n";
                print "Make";
                exit(0);
        }

        if(/Imakefile/){
#               printf STDERR "lang: $_ is Imakefile\n";
                print "Make";
                exit(0);
        }

        #Now check for extensions

        #get extension from filename
        ($file, $ext)=split(/\./);

    if ( "$lang{$ext}" eq "" ) {

                # no extension present, attempt guess using interpreter
                # (assuming it is the 1st line in the file-- we should really
                # look for "#!" magic rather than assume)
                $interp=`head -1 $_`;

                # parse out any flags that follow the interpreter name; e.g
                # "#!/usr/local/bin/perl -w"
                #--Robert Estes <estes@ece.ucdavis.edu>
                ($elm) = split ' ', $interp;  # First word, in case there are flags.
                $elm =~ s:.*/::;              # Interpreter

                #the last element should be the interpreter
                if( "$lang{$elm}" eq "" ){
                        print "unknown";
                        exit 1;
                }

                $ext=$elm;
        }

#       printf STDERR "lang: $_ is $lang{$ext}\n";
        print "$lang{$ext}";

}                                                       # for each file

exit 0;
```