```
#!/bin/bash

# output the contents of a file on the terminal.
# no. of arguments: 1
# Argument1: filename

mycat()
{
      cat $1
}

#
# print the following exactly as is from the lscpu output.
# CPU(s):                          8
# On-line CPU(s) list:             0-7
# Thread(s) per core:              2
# Core(s) per socket:              4
# Socket(s):                       1
#
# you could use any temporary file you want.
# just name it as prefixed with temp.lscpu.

lscpuinfo()
{
      lscpu | head -9 | tail -5
}

# given a file in argument 1, return '0' if
# the file is an C program executable else 1

is_exe()
{

      file $1 | grep executable > /dev/null
      return $?
}

# given a file in argument 1, return 0 if
# the file is an C program object file else 1

is_object()
{
      file $1 | grep "relocatable" > /dev/null
      return $?
}

# given a file in argument 1, return 0 if
# the file is an C program source file else 1

is_C()
{
      file $1 | grep "C source" > /dev/null
      return $?
}

# given a file in argument 1, return 0 if
# the file is a text  file else 1

is_text()
{
      file $1 | grep "ASCII text" > /dev/null
      return $?
}
```

```
# given a file in argument 1, return 0 if
# the file is an assembler source file 1 false

is_assembler()
{
      file $1 | grep "assembler" > /dev/null
      return $?
}

# Output the file names:type of file
# Total no. of arguments: 5
# Arguments 1-5 : All arguments are file names.
# output the file types of each of that file.
# Output should be of the following format.
# file1:assembler
# file2:executable
# file3:object
# file4:C
# file5:text
# Note the lack of any space inbetween.
# In the above , replace file1, file2.. etc to be
# the name given in the arguments.
# print "none" in the right column if you cannot find
# which kind of file it is or it is in a format that
# is other than what is listed above.

filetypes()
{
      for f in $@
      do
            found=0
            for k in exe assembler object C text
            do
                  is_$k $f
                  if [[ $? == 0 ]]
                  then
                        if [[ $k == exe ]]
                        then
                              echo "$f:executable"
                        else
                              echo "$f:$k"
                        fi
                        found=1
                        break
                  fi
            done
            if [[ $found != 1 ]]
            then
                  echo "$f:none"
            fi
      done
}
```