Problem 1 (5 points) Give Unix commands that perform the following tasks. (Each answer fits on a single line.) Assume that you execute them in your current directory, and they work in your current directory.

1. List all files in reverse order of the time when they are created (the most recent is last). `ls -lrt *`
2. List all files that contain the string "dummy".

   `grep dummy *`
3. Copy directory `dir1` (and all files in it) to `dir2`, where `dir2` does not exist.

   `cp -r dir1/ dir2`
4. Count the number of lines in all files. `wc *`
5. Set the permissions of all files such that you are the only user that can read and write.

   `chmod go-rw *`

Problem 2 (10 points) Write a shell script `findexec` that searches for a program (executable file) in your list of directories, that is, the directories stored in the path variable `$PATH`. For example, my path variable is set as

```
NN:~%echo $PATH
/Users/nednedialkov/bin:/sw/bin:/usr/sbin:/bin:/usr/bin:
/usr/local/bin:/usr/X11/bin:/usr/X11R6/bin:/usr/X11/bin
```

`findexec` searches for the file(s) given as argument(s) and outputs file location(s). For example

```
NN:~%findex gcc
/usr/bin/gcc
/usr/local/bin/gcc

NN:~%findex gnuplot '*plot' ls
/usr/local/bin/gnuplot
/sw/bin/pbmtoplot
/sw/bin/pic2plot
/sw/bin/plot
/sw/bin/tek2plot
/usr/local/bin/gnuplot
/bin/ls
```

You may find such a script useful. The Unix/Linux command `which` reports only the first command it finds, while this one reports all of them. For examples, `which gcc` gives on my system

```
NN:~%which gcc
/usr/bin/gcc
```

Solution

```
#!/bin/sh

for P in "$@"; do
  IFS=:
  for D in $PATH; do
    for F in $D/$P; do
      [ -x "$F" ] && echo $F
    done
  done
done
```

Problem 3 (10 points) Suppose that you need to create an executable with name `test` from three C programs stored in files `prog1.c`, `prog2.c`, and `prog3.c`. Write a makefile that creates `test`. Set a macro for the C compiler, a macro for the compiler flags, and also use the macros `$@` and `$?`. Include a target `clean`, so when `make clean` is typed, the object files and the executable are deleted.

Solution

```
OBJS=prog1.o prog2.o prog3.o
CC=gcc
CFLAGS=-Wall

test: $(OBJS)
        $(CC) $(LDFLAGS) -o $@ $?

clean:
        rm $(OBJS)
```

Problem 4 (10 points) Write an assembly program to find the minimum of

$$x^4 - 3x + 1$$

by stepping by one from $x = -2$ to $x = 2$.

Solution

```
include  "asm_io.inc"

segment .data
prompt1 db      "f(x) = ", 0
prompt2 db      "x = ", 0
res     db      "min_{x=-2,-1,0,1,2} f(x) = ", 0

segment .bss
minimum resd    1

segment .text
        global asm_main

asm_main:
        enter 0,0
        pusha
```

```
        mov ecx, 5
        mov byte[minimum], 0FFh

;       mov eax, [minimum]
;       call print_int
;       call print_nl

loop_start:
        mov eax, prompt2
        call print_string

        mov eax, 3
        sub eax, ecx    ; eax = 3-ecx
        mov edx, eax

        call print_int
        call print_nl
        mov eax, prompt1
        call print_string

        mov eax, edx
        mov ebx, eax    ; ebx = x
        imul ebx, eax   ; ebx = x^2
        imul ebx, eax   ; ebx = x^3
        sub ebx, 3      ; ebx = x^3-3
        imul ebx, eax   ; ebx = x^4-3x
        add ebx, 1      ; ebx = x^4-3x+1

        mov eax, ebx    ; eax = ebx

        call print_int ; print f(x)
        call print_nl
        call print_nl

        cmp [minimum], eax
        jl else_block
        mov [minimum], eax

else_block:

        loop loop_start

        mov eax, res
        call print_string
        mov eax, [minimum]
        call print_int
        call print_nl

        popa
        mov eax, 0
        leave
        ret
```