

NAME _____

(1) (10 points) Say how you would accomplish the following tasks.

(i) (a) Give ONE command to change your current directory from the ~poneil directory to your own cs240/hw2. (b) Give a command to list all files and subdirectories in the current directory, together with their protection. (c) Change the protection on a directory named "lookit" to delete permission for anyone other than yourself to look into or access files in it. (Use a command form where you only DELETE capabilities, rather than set all of them.)

Answer: (a)

(b)

(c)

(iii) Start the GDB debugger to run a program, and not yet running the program, with two breakpoints already set, (a) execute the program code until you get to the first breakpoint, (b) continue to execute until you get to the second breakpoint.

Answer: (a)

(b)

(2) (10 points) Say what is printed out by the following program. **DRAW PARENTHESES** to show order of evaluation in multi-term expression below to agree with page 53 of K&R. Show all intermediate work in **BINARY** of all expressions evaluated.

```
#include <stdio.h>
```

```
main()
{
    unsigned char x = '\165'
```

```
    x = x >> 2 & 0X1D ^ 023;
```

```
    printf ("octal x = %o, decimal 2*x = %d, hexadecimal 3*x = %x\n",
```

```
           x,
```

```
           2*x
```

```
           3*x );
```

```
}
```

Answer(PRECISE WORDING!):

cs240 Quiz 1, PRACTICE 1: Solutions

1 (i) (a) `cd ~/yourid/cs240/hw2` (or `cd ~/cs240/hw2`) (b) `ls -l` (or `-la`, even better) (c) `chmod og-rx lookit` (ii) (a) `r` or `run` (or `r <prog.in`) (b) `c`

(2) `x=(( x >> 2) & 0X1D) ^ 023;`

```
x = '\165' = 001 110 101    =    0111 0101
x = x >> 2;                =    0001 1101
                        0x1D  &    0001 1101
(x >> 2) & 0x1D            =    0001 1101
                        023 = 00 010 011    ^ 0001 0011
((x >> 2) & 0X1D) ^ 0123; =    0000 1110 = 00'001'110 = 016
```

`x = 016 = 14 (decimal), so 2*x 28;`

For hexadecimal `3*x`, mult by 11 bin:

```
00001110
X    11
-----
00001110
00001110 .
-----
000101010 = 0x2A
```

PRECISE TEXT: octal `x = 16`, decimal `2*x = 28`, hexadecimal `3*x = 2A`

Note. `printf` under `%o` gives 16, not 016, under `%x` gives 2A (or 2a) not 0x2A; all the output is on a single line.