

Problem 1

Suppose the file `foo.txt` contains the text "123456", `bar.txt` contains the text "abcdef", and `baz.txt` does not yet exist. Examine the following C code, and answer the questions below. (For space reasons, we are not checking error return codes, so assume that all functions return normally.)

```
int main() {
    int fd1, fd2, fd3, fd4, fd5, fd6;
    int status;
    pid_t pid;
    char c;
    /* foo.txt has "123456" */
    fd1 = open("foo.txt", O_RDONLY, 0);
    fd2 = open("foo.txt", O_RDONLY, 0);
    /* bar.txt has "abcdef" */
    fd3 = open("bar.txt", O_RDWR, 0);
    fd4 = open("bar.txt", O_RDWR, 0);
    /* baz.txt doesn't exist initially */
    fd5 = open("baz.txt", O_WRONLY | O_CREAT | O_TRUNC, S_IRUSR |
S_IWUSR); /* r/w */
    fd6 = dup(STDOUT_FILENO);
    dup2(fd5, STDOUT_FILENO);

    if ((pid = fork()) == 0) {
        dup2(fd3, fd2);
        read(fd3, &c, 1); printf("%c", c);
        write(fd4, "!@#$$%^", 6);
        read(fd3, &c, 1); printf("%c", c);
        read(fd1, &c, 1); printf("%c", c);
        read(fd2, &c, 1); printf("%c\n", c);
        exit(0);
    }
    wait(NULL);
    read(fd1, &c, 1); printf("%c", c);
    fflush(stdout);
    dup2(fd6, STDOUT_FILENO);
    printf("done.\n");
    return 0;
}
```

1. What will the contents of `baz.txt` be after the program completes?

a@1#

2

2. What will be printed on stdout?
done.

Problem 2

Modify the cpfile program in Figure 10.4 in ICS book so that it uses the Rio functions to copy standard input to standard output, MAXBUF bytes at a time.

```
#include "csapp.h"

int main(int argc, char *argv[]){
    int n;
    char buf[MAXBUF];
    while((n = Rio_readn(STDIN_FILENO, buf, MAXBUF)) != 0)
        Rio_writen(STDOUT_FILENO, buf, n);
    exit(0);
}
```

Problem 3

Write a program to implement the function of "ls -l > foo.txt" that causes the shell to load and execute the ls program, with standard output redirected to disk file foo.txt. You don't need to implement the function 'ls' and suppose the path of 'ls' is '/bin/ls'.

```
#include "csapp.h"

int main(int argc, char *argv[]){
    if(argc != 2){
        error("Usage: %s -l\n", argv[0]);
        exit(-1);
    }
    pid_t pid;
    if((pid = fork()) == 0){
        int fd = open("foo.txt", O_WRONLY|O_CREAT);
        if(fd < 0){
            exit(-1);
        }
        dup2(fd, 1);
        close(fd);
        execve("/bin/ls", argv, NULL);
    }else{
        waitpid(-1, NULL, 0);
    }
}
```

```
    exit(0);  
}
```

Problem 4

Write a program to implement the function “echo” just like function “echo” executed in Unix system.

```
#include <stdio.h>  
#include <string.h>  
int main(int argc, char *argv[]){  
    if(argc == 1){  
        printf("\n");  
    }else{  
        for(int i = 1; i < argc; i++){  
            printf("%s%c", argv[i], (i < argc - 1) ? ' ' : '\n');  
        }  
    }  
    return 0;  
}
```