EECE.4810/EECE.5730 Spring 2019: Exam 1
Reference Material

Question 1b, 1c, 1d (definitions for programs *pr1* and *pr2*)

pr1:

```
int var1;

int main(int argc, char **argv) {
   int var2 = 5730;
   char str[15];
   pid_t pid, pid2;

   var1 = atoi(argv[1]);

   pid = fork();        (1)
   if (pid == 0) {
      printf("P1 child 1: %d %d\n",
               var1, var2);
      var1 = var2;

      pid2 = fork();    (2)
      if (pid2 == 0)
         printf("P1 child 2: %d %d\n",
                  var1, var2);
      else if (pid2 > 0) {
         wait(NULL);
         sprintf(str, "%d %d",
                  var1, var2);
         execlp("./pr2", "pr2",
                  str, NULL);
      }
   }
   else if (pid > 0) {
      var2 = var1;
      wait(NULL);
      printf("P1: %d %d\n",
               var1, var2);
   }
   return 0;
}
```

pr2:

```
int main(int argc, char **argv) {
   pid_t pid;
   int var3, var4;

   printf("P2: %s\n", argv[1]);
   sscanf(argv[1], "%d %d",
           &var3, &var4);
   var4 = var4 - 920;

   pid = fork();       (3)
   if (pid > 0) {
      wait(NULL);
      printf("P2 parent: %d %d\n",
               var3, var4);
   }
   else if (pid == 0)
      printf("P2 child: %d %d\n",
               var3, var4);
   return 0;
}
```

Notes:

- ***pr1*** always runs first and is invoked with the command line: `./pr1 4810`
- These programs use the following functions that may be unfamiliar:
  - `atoi()`: Converts its string argument to an integer—so, for example, `atoi("10") = 10`
  - `sprintf()`: Behaves similarly to `printf()`/`fprintf()`, but first argument is a character array to which a string is printed.
  - `sscanf()`: Behaves similarly to `scanf()`/`fscanf()`, but first argument is a string from which inputs are read.