Assignment 2
Due: January 19, 1-00 pm

1. Write a C program that will search an array of integers for another given integer. However, to speedup the search, the search is done in parallel by two child processes. The parent process reads in the number of integers (max. 100), and then the integers in an array. It also reads in the integer to be searched. It then creates two child processes. The first child process searches the first half of the array, and the second child process searches the second half. If the integer is found, its index in the array is sent to the parent through a pipe. If it is not found, a -1 is sent to the parent through a pipe. The parent waits for both child processes to finish and then prints an appropriate message.

    Name your program *search.c*.

2. In this assignment, we will complete the shell that we started writing in Assignment 1. Start with the program in Assignment 1. In addition to the features it already has, your program should do the following:
    o Supports the **builtin** command "cp " (usual Linux command to copy files, no option of cp need to be supported. However, the flenames may contain a full pathname)
    o Supports *background* execution of commands. Normally when you type a command at the shell prompt, the prompt does not return until the command is finished. For background executions, the prompt returns immediately, the command continues execution in the background. Typing an "&" at the end of a command (for ex., a.out&) should make it execute in the background
    o Should be able to redirect the output of a program to a file using ">" and read the input of a program from a file using "<". For example, typing "a.out > outfile" should send whatever was supposed to be displayed on the screen by a.out to the file *outfile* . Similarly, typing "a.out < infile" should make a.out take the inputs from the file *infile* instead of the keyboard.
    o Should be able to redirect the output of one command to the input of another by using the "|" symbol. For example, if there is a program *a.out* that writes a string "abcde" to the display, and there is a program *b.out* that takes as input a string typed from the keyboard, counts the number of characters in the string, an displays it, then typing "a.out | b.out" at your shell prompt should display 5 (the output "abcde" from *a.out* was fed as input to *b.out* , and 5, the number of characters in "abcde", is printed). Use the pipe command. Any number of redirections upto a maximum of 4 redirections should be allowed (for ex., a | b | c, a| b | c| d |e,…).

As before, to run your shell, write another C program that will create a child process and call an appropriate form of exec to run the program above. The parent process simply waits for the child to finish (execute the "exit" command), after which it also exits.

Name the file *shell.c* as before.

Follow instructions on the submission site to submit your files.