Worksheet: Pipe (workers & master)

In the last worksheet we wrote a program that forked one child for each command line argument. The child computed the length of the command line argument and exits with that integer as the return value. The parent sums these return codes and reports the total length of all the command line arguments together.

For this worksheet, write a similar program where each child communicates the length to the parent through a pipe rather than via a return code.

```
int main(int argc, char **argv) {
  // Declare any variables you need
  int pipe_fd[argc][2];//shoule be argc-1

  ___
  // write the code to loop over the command line arguments
  for (int i = 1; i < argc; i++) {
    // Before we call fork.  Call pipe.
    pipe(pipe_fd[i]);//we can use the name of the array as a pointer to an array, also do error checking

    ___
    int result = fork();
    if (result < 0) {
      perror("fork");
      exit(1);
    } else if (result == 0) {  // child process
      // child only writes to the pipe so close reading end
      for ( int j = 1;  j <= i;  j++ ) {
      ___close(pipe_fd[i][0]);
      }
      // before we forked the parent had open the reading ends to
      // all previously forked children -- so close those
      int len = strlen( argv[i] );
      write(pipe_fd[i][1], &len, sizeof(int) );//also some error checking
      close(pipe_fd[i][1]);//also some error cheking
      exit(0);
    } else {   //in the parent
      // in the parent but before doing the next loop iteration
      close(pipe_fd [i][1] );

      ___

      ___
    }
  }
```

```c

  // Only the parent gets here
  int sum = 0;
  // read one integer from each child
  // for now print it out as well as adding it in
  int contribution;
  for (int i=1; i< argc; i++) {
    if (read(pipe_fd[i][0], &contribution, sizeof(int)) == -1) {
      perror("reading from pipe from a child");
    }
    fprintf(stderr, "I just read a %d from child %d\n",
   contribution, i);
    sum += contribution;
  }

  printf("The length of all the args is %d\n",sum);
  return 0;
}
```