1. Drew Sadler

```
//input
#include <stdio.h>
#include <unistd.h>


int main( int argc, char* argv[] )
{
while(1)
{
        int buffer=1024;
        char message[buffer];
        int length = sizeof(message)/sizeof(message[0]);
        char  *input_string=fgets(message,length,stdin);
        if(input_string==NULL){
                break;
        }
        else{
                printf("Program 2 got: %s\n", input_string);
        }
}

}

//output
// Drew Sadler
// Include file goes here
#include<stdio.h>
int main( int argc, char* argv[] ){

//Make a call to fprintf() here
fprintf(stdout,"Hello, world!\n");

return 0;
}
```

2.
```
[asadler1@hopper3 Studio_07]$ ./pipe
Program 2 got: Hello, world!

hello
a
^C
```
3. *pipefd[0]* refers to the read end of the pipe. *pipefd[1]* refers to the write end of the pipe
4. *Oldfd is closed after a successful run.*

5.
```
dup2(pipefd[1], STDOUT_FILENO);
dup2(pipefd[0], STDIN_FILENO);
```
6.
```
[asadler1@hopper3 Studio_07]$ ./pipe
Program 2 got: Hello, world!

hello
aj
a
lsdjfk
^C
```
7.
```
[asadler1@hopper3 Studio_07]$ ./pipe
Program 2 got: Hello, world!
```
8. *It allows for very dynamic programming and manipulation of data, as it can traverse and edit any file you may be wanting to pipe/connect and extract or retrieve data from any type of source.*
   Can also allow for chains of programs to be piped together and have different files accessed or executed, making large programs easy to communicate with each sub-process of it and can trace the data through each connection easily, grabbing and placing whatever you need wherever you want it to be streamed to