Assignment3
Sanchit Trivedi
2018091

“/bin/ls | /bin/sort | /bin/uniq”
For the above command my code will first split at the pipe symbol and treat the obtained strings as individual commands. For each command redirection will be handled if any. Also my code maintains a count of the pipe symbol;

```
void call_pipe(char **in, int *pcount, int i){
   if(i == *pcount - 1){
        //if it is the final process
        char cpy[MAXCOMMANDS];
        strcpy(cpy, in[i]); //copy of the cmd

        char *args[MAXCOMMANDS];
                Here we will tokenize the str by splitting at whitespaces;
        int x;
        if((x = redirectionCheck(in[i])) < 0){//check for redirections
                        execute(args);//If no simply execute
        }
        else{
                redirection(in[i], x);//else redirect
                return 1;
        }
   }
   if(i < *pcount){// not final process
        int fd[2];//pipe array
        pid_t pid;
        char cpy[MAXCOMMANDS];//cpy for manipulation
        char *args[MAXCOMMANDS];

        if(pipe(fd) < 0){
                printf("pipe failed");
                exit(1);
        }
        pid = fork()
        if(pid < 0){
                printf("fork failed");
                exit(1);
```

```
        }

        if(pid != 0){
                dup2(fd[1], 1);
                close(fd[0]);
                in[i+1] = NULL;

                strcpy(cpy,in[i]);

               //tokenization
                int x;
                if((x = redirectionCheck(in[i])) < 0){
                        execute(args);
                }
                else{
                        redirection(in[i], x);}
                wait(NULL);//wait for child process

        }
        else{
                if(i != *pcount-1){
                        dup2(fd[0], 0);
                }
                close(fd[1]);

                i++;
                call_pipe(in, pcount, i);
        }
    }
}
```