```
/*
* Course: CS 100 Fall 2013
*
* First Name: Luis
* Last Name: Garcia
* Username: lgarc018
* email address: lgarc018@ucr.edu
*
*
* Assignment: HW5
*
* I hereby certify that the contents of this file represent
* my own original individual work. Nowhere herein is there
* code from any outside resources such as another individual,
* a website, or publishings unless specifically designated as
* permissible by the instructor or TA.
*/
#include <iostream>
#include <vector>
#include <sys/types.h>
#include <sys/wait.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <iostream>

using namespace std;

//im putting my commands into a vector
//so read in the linea and vectorize
vector<string> vectorize(string input)
{
    string push;
    vector<string> retVector;
    int i;
    for(i=0; i<input.size(); i++)
    {
        if(input[i] != ' ')
        {
            push+=input[i];
        }
        else
        {
            if(push == "")
                continue;
            retVector.push_back(push);
            push = "";
        }
    }
    if(push != "")
        retVector.push_back(push);

    return retVector;
}


int execute (vector<string> vectArgv)
{

    int status;
```

```
int pid = fork();
const char **argv = new const char* [vectArgv.size()];
const char *program = vectArgv[0].c_str();
int i;
string token;
vector<string> commands;
//int pipefd[2];
//pipe(pipefd);
switch (pid)
{
    case -1:
        cout << "forked up!\n";
        return -1;
    case 0:
        for (i=0; i<vectArgv.size() && vectArgv[i]!="|"; i++)
        {
            if(vectArgv[i] != ">" && vectArgv[i] != "<")
            {
                argv[i] = vectArgv[i].c_str();
            }
            else
            {
                token = vectArgv[i];
                if(i+1 == vectArgv.size())
                {
                    //if(token != "|")
                        cerr << "Error expected file namae!\n";
                    //else
                      //  cerr << "Open pipe!\n";
                    _exit(1);
                }else{
                    if(token == ">")
                    {
                        int output;
                        output = open(vectArgv[i+1].c_str(), O_WRONLY
                                      | O_TRUNC | O_CREAT, S_IRUSR|S_IWUSR);
                        dup2(output, 1);
                        close(output);
                    }else{
                        int input;
                        input = open(vectArgv[i+1].c_str(), O_RDONLY);
                        dup2(input,0);
                        close(input);
                    }
                }
                argv[i] = NULL;
            }
        }

        argv[i++] = NULL;



        if(vectArgv[i-1] == "|")
        {
            cerr << "If this worked it would have piped here...";
            int fds[2];
            pipe(fds);
            int pid2;
            if(fork() == 0)
            {
            dup2(fds[0],0);
            close(fds[1]);
```

```
            vector <string> newArgv;
               //onst char **argv = new const char* [vectArgv.size()];
               //cout << endl << endl;
               for(i++;i<vectArgv.size(); i++)
               {
                    //cout << vectArgv[i] << endl;
                    newArgv.push_back(vectArgv[i]);
               }
               execute(newArgv);

            }else if((pid2 = fork()) == 0)
            {
              dup2(fds[1], 1);
              close(fds[0]);
              if(execvp(program, (char**)argv)==-1)
                    perror("execvp failed");
            }
            else
            waitpid(pid2, NULL,0);
            exit(1);
        }

        else{


         execvp(program, (char **)argv);
//          exit(1);
        }



    default:
        int status;
        return waitpid(-1, NULL, 0);
}
}

int main()
{
    string input;
    vector <string> argv;
    while(true)
    {
        cout << "~badshell% ";
        getline(cin, input);
        argv = vectorize(input);

        /*for(int i=0; i<argv.size(); i++)
        {
            cout << argv[i] << " " << i << endl;
        }
        */
        if (execute(argv) < 0)
            perror(argv[0].c_str());

        input = "";
        argv.clear();
    }
}
```