OS LAB EXERCISE 2

Mahmoud Elshenawy 900183926

The submission contains two main files:

- **makefile:**
  - this is to build the intermediate stages of compiling and contains a ‘run’ target to directly run the program by typing ‘make run’

- **shell.c:** It contains some functions:
  - ***read_command***: takes the command from the user
  - ***build_args***: it parses the command taken from the user and store it in ‘argv’
  - ***set_program_path***: it updates the path parameter to the correct path needed by execve to execute different programs.
  - ***find_char***: it searches ‘argv’ for certain character and updates the ‘arr’ if it found the character and returns how many times the character apperead.
  - ***Handle_redirection***: it takes argv then creates an updated version of it and then duplicates the suitable file for redirection and returns a flag indicating if there is redirection
  - ***create_subargument***: it takes the argv and then parses it for piping cases.

- **Main:**
  - The main contains an infinite loop to consistently take commands unless he/she types exit, in which case, the loop is broken.
  - Inside the loop, we check for certain commands first, such as ‘cd, echo , x=?’. If we don’t find it, we see if there is a pipe there, we execute the corresponding the pipe. Inside we create another loop to go through the commands one by one creating the pipes needed and checking for redirection for each command and then executes.
  - If it doesn’t happen to be any of the previous commands, then we check only for redirection, by calling handle_redirection funtion, and then executes the parsed command or the original depending on if a redirection happens or not.

Data Structures:

- Pointer to a pointer of characters
- 2d dynamic array for pipes
- 1d dynamic array for searching