**Programmer Manual**

**How to run the code:**

* Get a copy of the following files:
  + **proj2.ASM**,
  + **iofar.lib,**
  + **compile.bat**
* Save the files to the following drive path: **C:\tasm\tasm\cs221\p2**
* Click on the **DOS box** icon and watch the console open.
* At the DOS box prompt, make sure to change directory to your **P2** file with the specified path listed.
  + Console should now look like this:

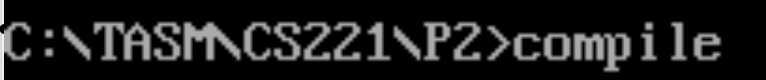
A picture containing food, drawing

Description automatically generated

* To Assemble and Link the source file, type the following line:

**compile**

* + Console should now look like this:



* **Proj2.EXE is generated,** to run the program at the prompt type: **Proj2**

**When the program executes:** The program will prompt the user for:

- a number of disks: **num** (range of num is 1, 2 or 3.)

- an starting pole position: **pStart** (ranges from 3 up to 7)

- an ending pole position: **pFinal** (ranges from 3 up to 7, but not the same as pStart)

**Major Procedures used:**

* **H procedure (recursive procedure)**
  + **void H(counter, num, pStart, pFinal);**
* **Greet procedure**
  + **void Greet();**

**Structure of the Code:**

* Enter **Greet** **PROC** for sending greeting message.
* Receive Variables and do error checking and error recovery.
* Enter **H PROC** for solving Towers of Hanoi problem while printing proper movements.

**Functionality of certain aspects of the code as to the project requirements:**

* It will prompt user for inputting 3 variables: **num**, **pStart** and **pFinal**;
* For error checking and recovery of the three inputs, **WHILE01**, **WHILE02**, and **WHILE03** loops are used.
* Variable **counter** is initialized as 0 for counting the total number of movements;
* The recursiveprocedure **H(counter, num, pStart, pFinal)** is used for solving the tower puzzle recursively.
* Sudo Code of H procedure:

if (n == 1) then print(counter, “move disk “,n, “ from pole “,r, “ to pole ”, s)

else

{

H(counter, n-1, s, 6-s-f)

print (counter, “move disk “, n, “ from pole “, r, “to pole “, s)

H(counter, n-1, 6-s-f, s)

}

endif

return // end of procedure