

Splitting functions(modules) to seperate files

step1
create a header file game.h and cut paste all the declarations from game.c to game.h

step2
for each function definition create a seperate source file and cut paste the function definition from game.c to the respective source file and include "game.h" and stdio.h or other header files as required.

e.g create displayMenu.c source file and cut paste the displayMenu function definition from game.c to displayMenu.c and include game.h header file (since it contains the function declaration for displayMenu) and stdio.h (since it contains the function declarations for printf and scanf).

step3
after repeating step2 for all the functions present in game.c, the only function left in game.c is the 'main' function.

In a nutshell, function declarations go to header file, function definitions go to a seperate source file and function calls will either be there in the source file containing main function as well as in some function definitions (like game() function which calls calculateScore, displayHint etc).

Compilation procedure.

step1. compiling individual modules

```
gcc -c displayMenu.c -Wall
gcc -c getOption.c -Wall
gcc -c guess.c -std=c99 -Wall
gcc -c processOption.c -Wall
gcc -c generateRandomNumber.c -Wall
gcc -c getGuessedNumber.c -Wall
gcc -c displayHint.c -Wall
gcc -c calculateScore.c -Wall
gcc -c getPlayerDetails.c -Wall
gcc -c printPlayerDetails.c -Wall
```

step2. Linking all the individual modules and creating an executable

```
gcc -o game.o game.c displayMenu.o getOption.o guess.o processOption.o generateRandomNumber.o
getGuessedNumber.o displayHint.o calculateScore.o getPlayerDetails.o printPlayerDetails.o -Wall -
std=c99
```

Running the program
./game.o

Note:
If any changes are made to individual module then it must be recompiled and also step2 must be repeated.