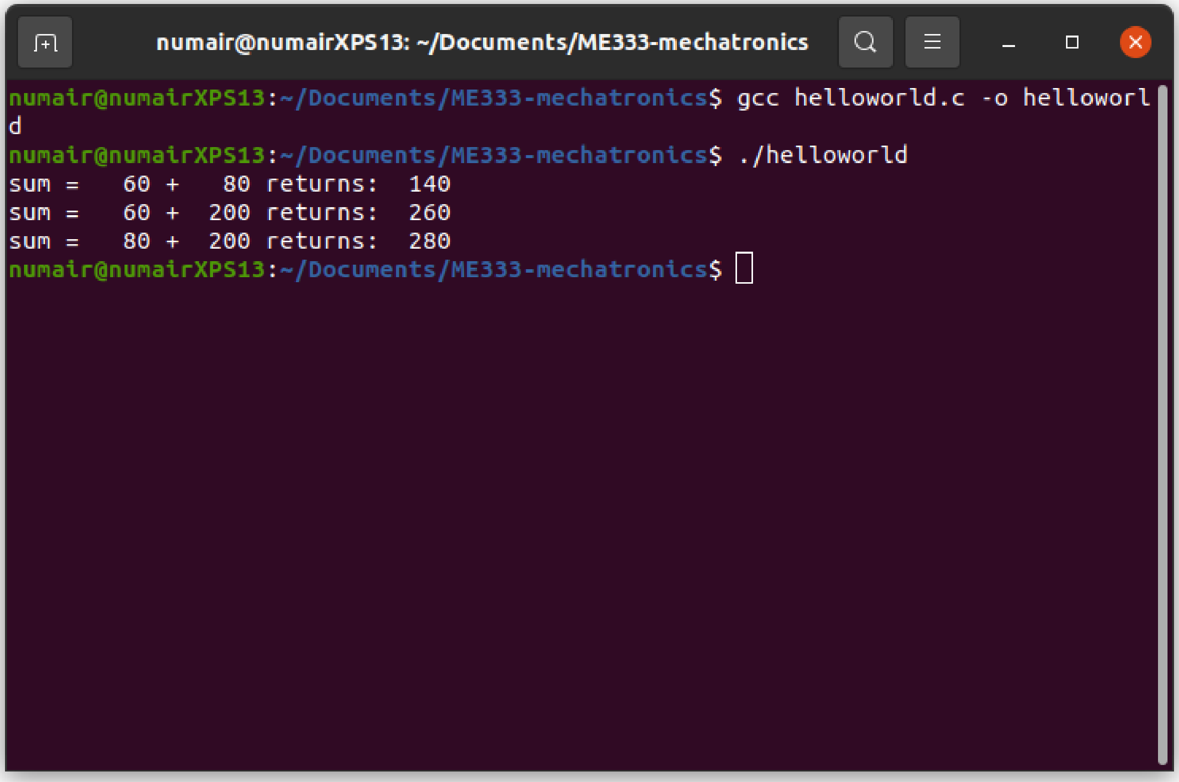
Homework 1
ME 333 - W2021
Numair Ahmed

18. The four steps of the gcc command: preprocessing, compiling, assembling, and linking
outputs:

- preprocessing: compresses input source code to remove comments and expands #include files. output is a temporary .i file.
- compiling: takes the .i file as input, checks for C syntax errors, and converts the .i file to assembly code. Output is a .s file.
- assembling: translate the .s file to machine code and generate a .o file to be used by the linking step
- linking: links the functions with the intended definition from source. The output is the final executable file

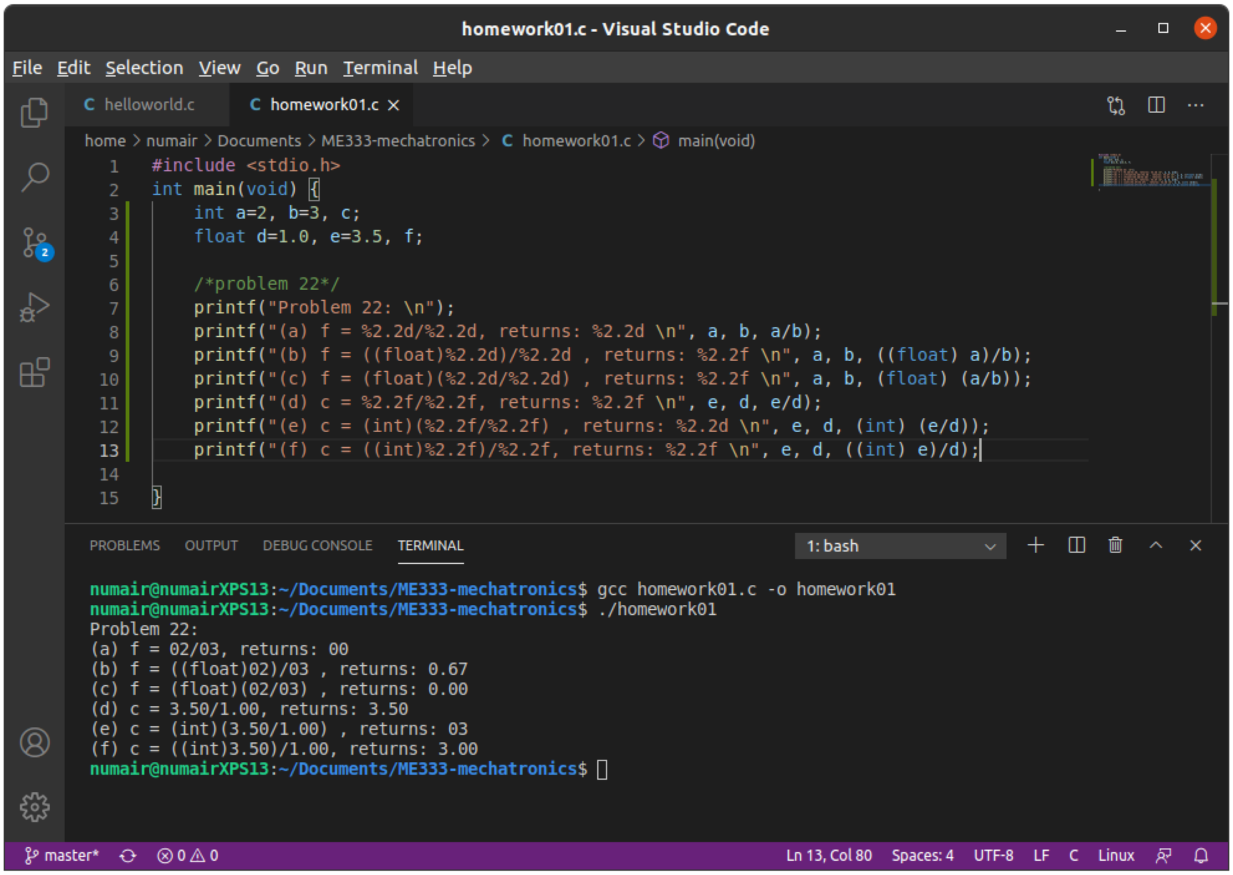
19. In the case of "int main(void) { }", the return type is indicated by "int" and typically returns a value of 0 upon successful completion of the function.

21.



```
numair@numairXPS13:~/Documents/ME333-mechatronics$ gcc helloworld.c -o helloworl
d
numair@numairXPS13:~/Documents/ME333-mechatronics$ ./helloworld
sum =   60 +   80 returns:  140
sum =   60 +  200 returns:  260
sum =   80 +  200 returns:  280
numair@numairXPS13:~/Documents/ME333-mechatronics$ 
```

22.


homework01.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
helloworld.c
homework01.c
home > numair > Documents > ME333-mechatronics > homework01.c > main(void)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash
master*
0 0
Ln 13, Col 80 Spaces: 4 UTF-8 LF C Linux

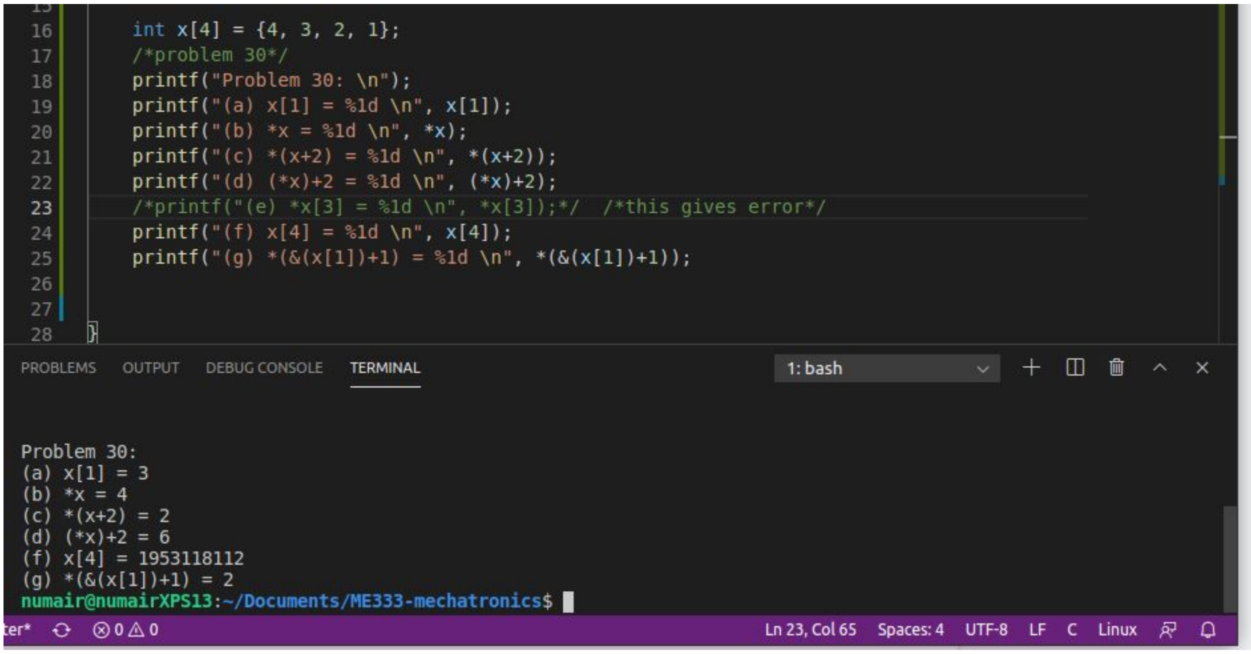
```c
#include <stdio.h>
int main(void) {
    int a=2, b=3, c;
    float d=1.0, e=3.5, f;

    /*problem 22*/
    printf("Problem 22: \n");
    printf("(a) f = %2.2d/%2.2d, returns: %2.2d \n", a, b, a/b);
    printf("(b) f = ((float)%2.2d)/%2.2d , returns: %2.2f \n", a, b, ((float) a)/b);
    printf("(c) f = (float)(%2.2d/%2.2d) , returns: %2.2f \n", a, b, (float) (a/b));
    printf("(d) c = %2.2f/%2.2f, returns: %2.2f \n", e, d, e/d);
    printf("(e) c = (int)(%2.2f/%2.2f) , returns: %2.2d \n", e, d, (int) (e/d));
    printf("(f) c = ((int)%2.2f)/%2.2f, returns: %2.2f \n", e, d, ((int) e)/d);

}
```

```
numair@numairXPS13:~/Documents/ME333-mechatronics$ gcc homework01.c -o homework01
numair@numairXPS13:~/Documents/ME333-mechatronics$ ./homework01
Problem 22:
(a) f = 02/03, returns: 00
(b) f = ((float)02)/03 , returns: 0.67
(c) f = (float)(02/03) , returns: 0.00
(d) c = 3.50/1.00, returns: 3.50
(e) c = (int)(3.50/1.00) , returns: 03
(f) c = ((int)3.50)/1.00, returns: 3.00
numair@numairXPS13:~/Documents/ME333-mechatronics$ 
```

27.
- Take advantage of the modularity of the program. Test individual functions for correct behavior given an input and known expected output.
- Once a suspect module is narrowed down, put in print statements at lines in the code that seem suspect.

28.
see modified invest.c file submitted to Canvas

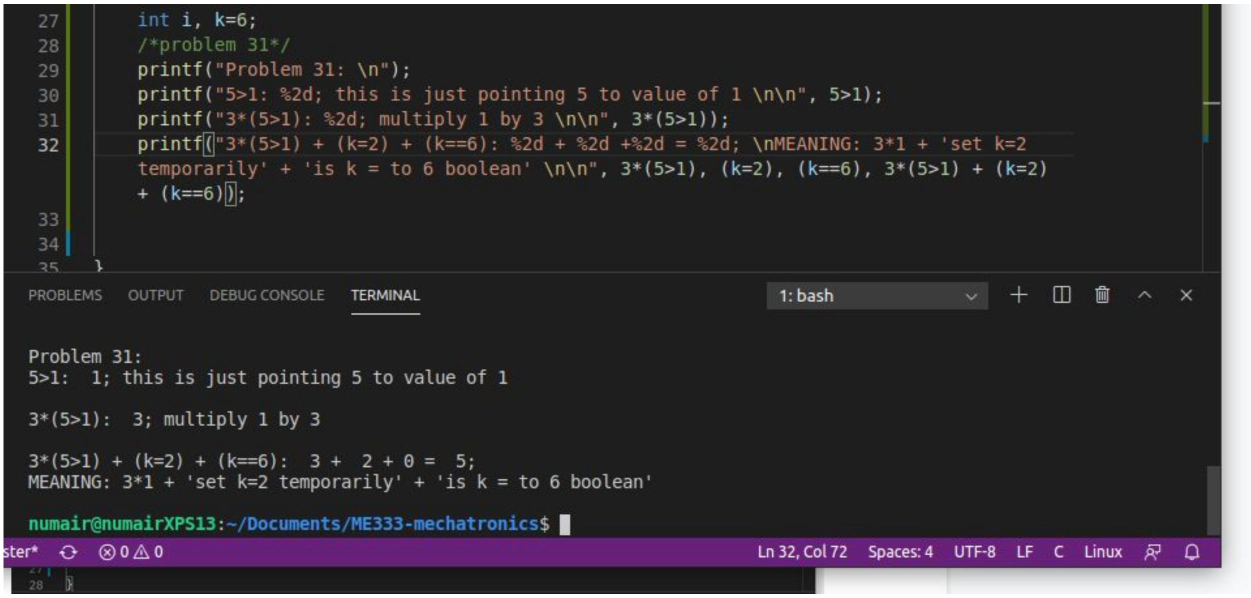
30.



```
    int x[4] = {4, 3, 2, 1};
    /*problem 30*/
    printf("Problem 30: \n");
    printf("(a) x[1] = %1d \n", x[1]);
    printf("(b) *x = %1d \n", *x);
    printf("(c) *(x+2) = %1d \n", *(x+2));
    printf("(d) (*x)+2 = %1d \n", (*x)+2);
    /*printf("(e) *x[3] = %1d \n", *x[3]);*/  /*this gives error*/
    printf("(f) x[4] = %1d \n", x[4]);
    printf("(g) *(&(x[1])+1) = %1d \n", *(&(x[1])+1));


}
```

```
Problem 30:
(a) x[1] = 3
(b) *x = 4
(c) *(x+2) = 2
(d) (*x)+2 = 6
(f) x[4] = 1953118112
(g) *(&(x[1])+1) = 2
numair@numairXPS13:~/Documents/ME333-mechatronics$
```

31.



```
    int i, k=6;
    /*problem 31*/
    printf("Problem 31: \n");
    printf("5>1: %2d; this is just pointing 5 to value of 1 \n\n", 5>1);
    printf("3*(5>1): %2d; multiply 1 by 3 \n\n", 3*(5>1));
    printf("3*(5>1) + (k=2) + (k==6): %2d + %2d +%2d = %2d; \nMEANING: 3*1 + 'set k=2
    temporarily' + 'is k = to 6 boolean' \n\n", 3*(5>1), (k=2), (k==6), 3*(5>1) + (k=2)
    + (k==6));


}
```

```
Problem 31:
5>1:  1; this is just pointing 5 to value of 1

3*(5>1):  3; multiply 1 by 3

3*(5>1) + (k=2) + (k==6):  3 +  2 + 0 =  5;
MEANING: 3*1 + 'set k=2 temporarily' + 'is k = to 6 boolean'

numair@numairXPS13:~/Documents/ME333-mechatronics$
```