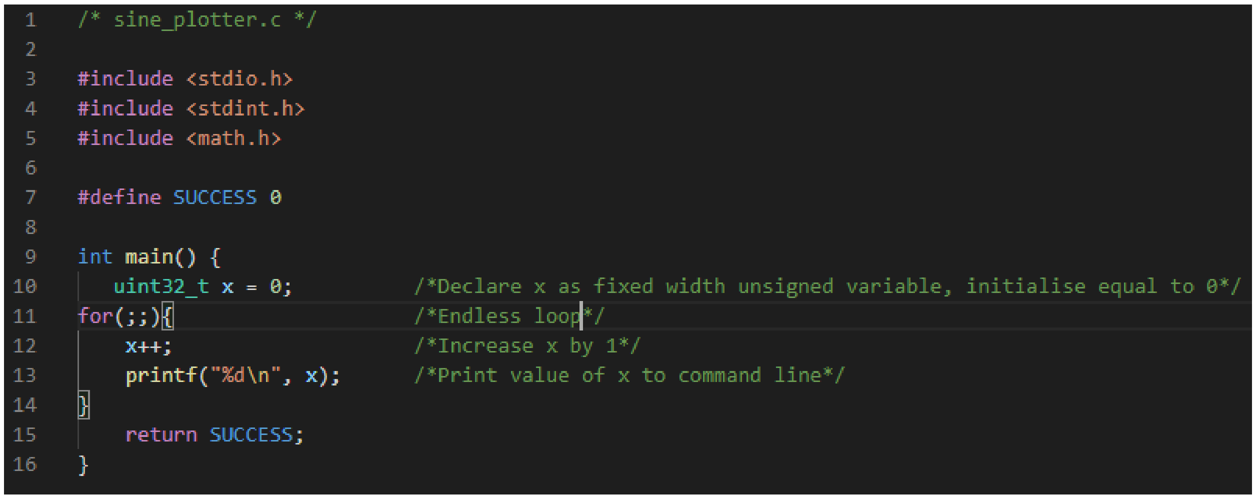
14/10/20

I started my code by having my loop print the values of x on consecutive lines.



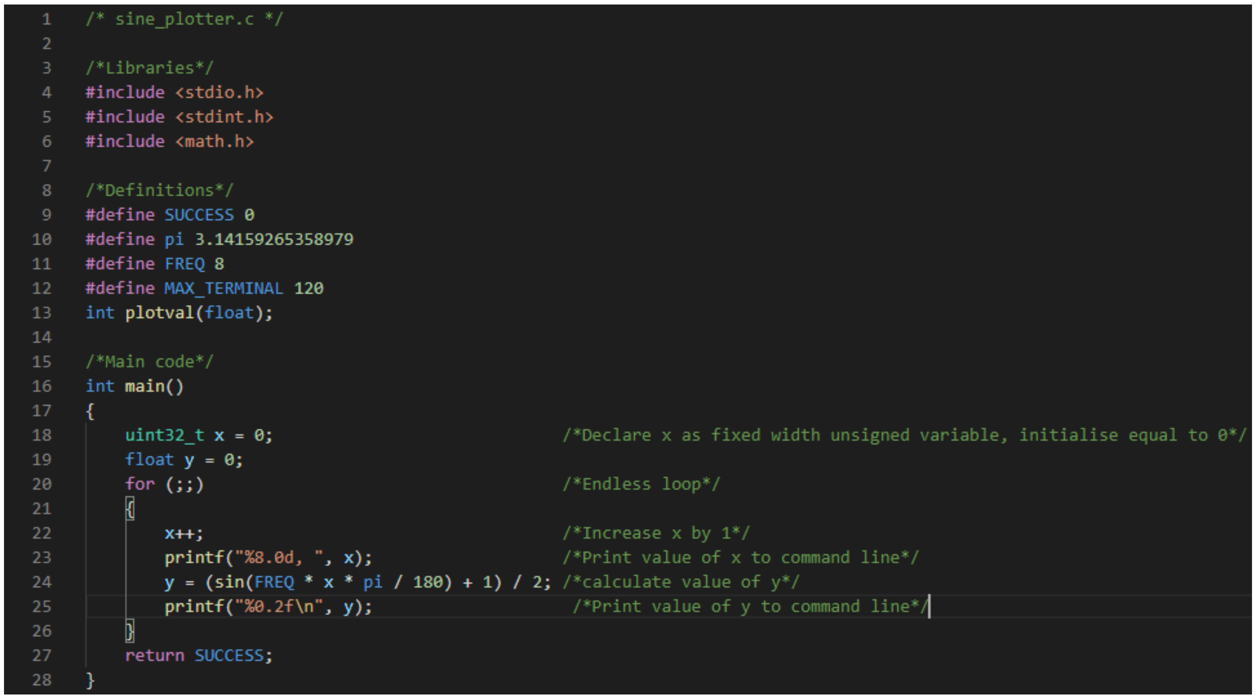
```c
/* sine_plotter.c */

#include <stdio.h>
#include <stdint.h>
#include <math.h>

#define SUCCESS 0

int main() {
   uint32_t x = 0;          /*Declare x as fixed width unsigned variable, initialise equal to 0*/
for(;;){                    /*Endless loop*/
    x++;                    /*Increase x by 1*/
    printf("%d\n", x);      /*Print value of x to command line*/
}
    return SUCCESS;
}
```

I then had my program calculate the value of y, being the sine of x. And then translate the value of y to the range 0 and 1.



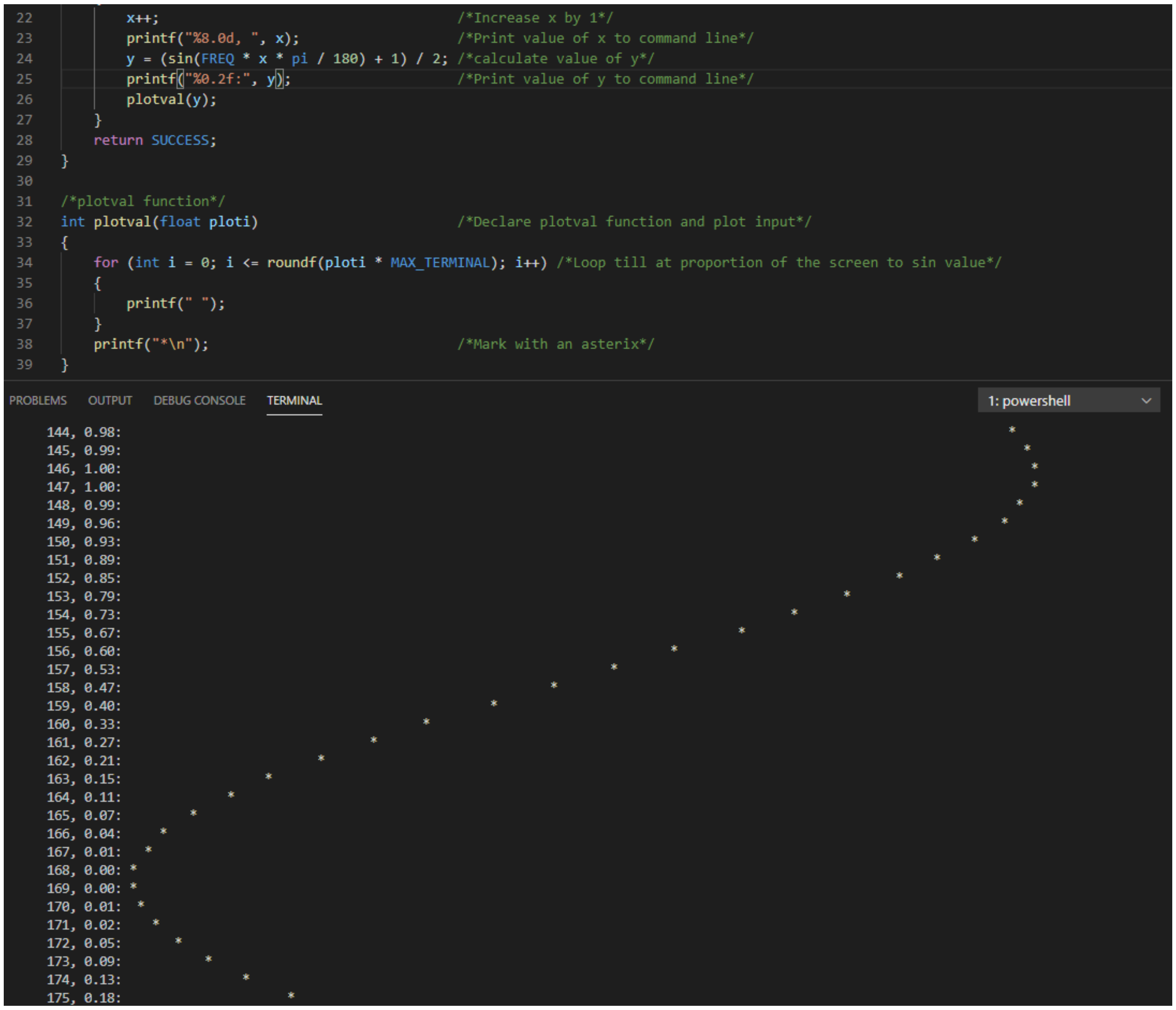
```c
/* sine_plotter.c */

/*Libraries*/
#include <stdio.h>
#include <stdint.h>
#include <math.h>

/*Definitions*/
#define SUCCESS 0
#define pi 3.14159265358979
#define FREQ 8
#define MAX_TERMINAL 120
int plotval(float);

/*Main code*/
int main()
{
    uint32_t x = 0;                         /*Declare x as fixed width unsigned variable, initialise equal to 0*/
    float y = 0;
    for (;;)                                /*Endless loop*/
    {
        x++;                                /*Increase x by 1*/
        printf("%8.0d, ", x);               /*Print value of x to command line*/
        y = (sin(FREQ * x * pi / 180) + 1) / 2; /*calculate value of y*/
        printf("%0.2f\n", y);                /*Print value of y to command line*/
    }
    return SUCCESS;
}
```

I then added the plotval() function that I made during my preparation for this lab. This created a trace of the sine function on the terminal.

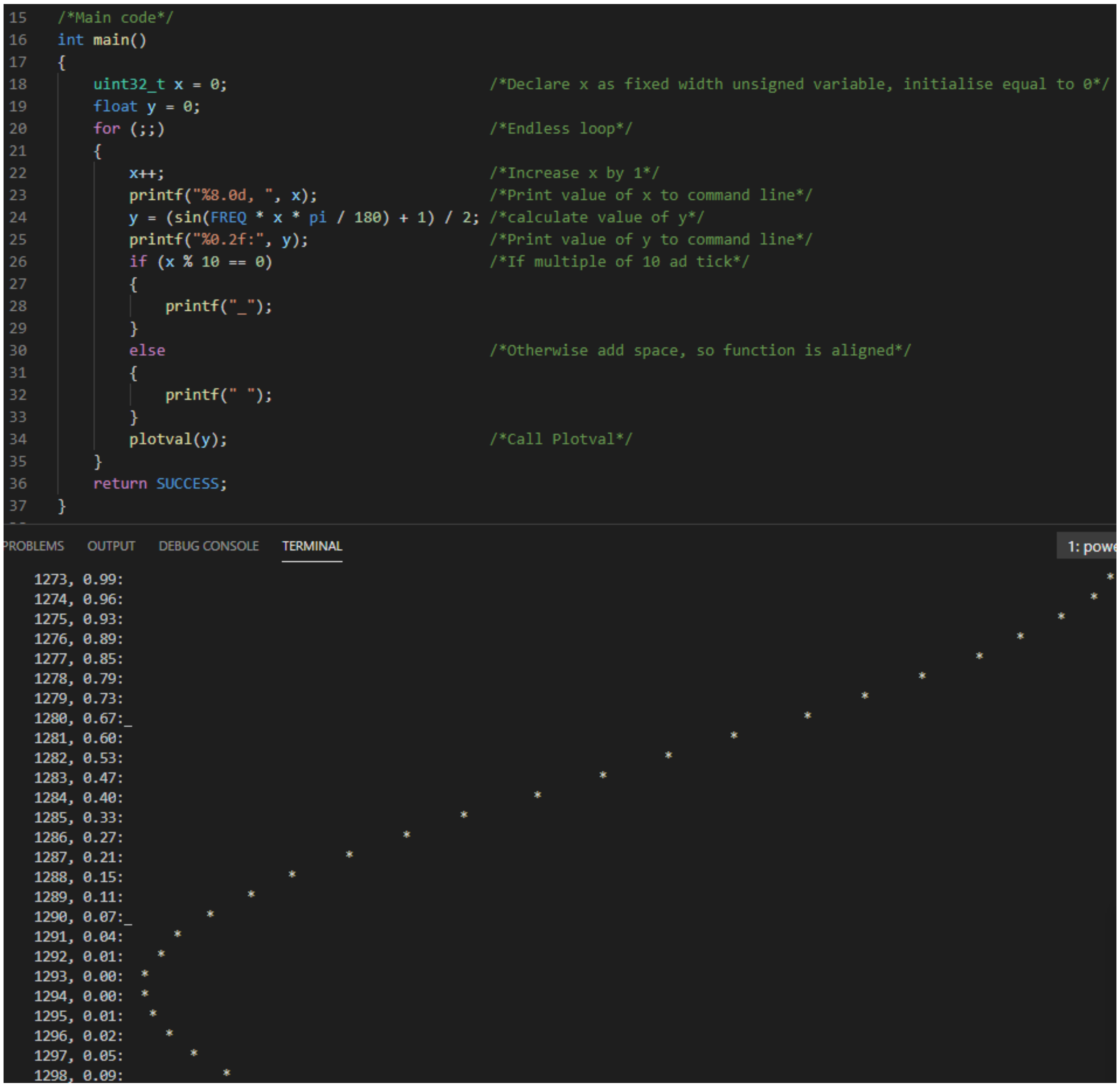


```c
        x++;                                    /*Increase x by 1*/
        printf("%8.0d, ", x);                   /*Print value of x to command line*/
        y = (sin(FREQ * x * pi / 180) + 1) / 2; /*calculate value of y*/
        printf("%0.2f:", y);                    /*Print value of y to command line*/
        plotval(y);
    }
    return SUCCESS;
}

/*plotval function*/
int plotval(float ploti)                        /*Declare plotval function and plot input*/
{
    for (int i = 0; i <= roundf(ploti * MAX_TERMINAL); i++) /*Loop till at proportion of the screen to sin value*/
    {
        printf(" ");
    }
    printf("*\n");                              /*Mark with an asterix*/
}
```

```
     144, 0.98:                                                                                                                       *
     145, 0.99:                                                                                                                        *
     146, 1.00:                                                                                                                         *
     147, 1.00:                                                                                                                         *
     148, 0.99:                                                                                                                        *
     149, 0.96:                                                                                                                    *
     150, 0.93:                                                                                                                 *
     151, 0.89:                                                                                                            *
     152, 0.85:                                                                                                       *
     153, 0.79:                                                                                                *
     154, 0.73:                                                                                         *
     155, 0.67:                                                                                 *
     156, 0.60:                                                                         *
     157, 0.53:                                                                 *
     158, 0.47:                                                         *
     159, 0.40:                                                 *
     160, 0.33:                                         *
     161, 0.27:                                 *
     162, 0.21:                          *
     163, 0.15:                   *
     164, 0.11:              *
     165, 0.07:         *
     166, 0.04:      *
     167, 0.01:  *
     168, 0.00: *
     169, 0.00: *
     170, 0.01:  *
     171, 0.02:   *
     172, 0.05:       *
     173, 0.09:            *
     174, 0.13:                 *
     175, 0.18:                       *
```

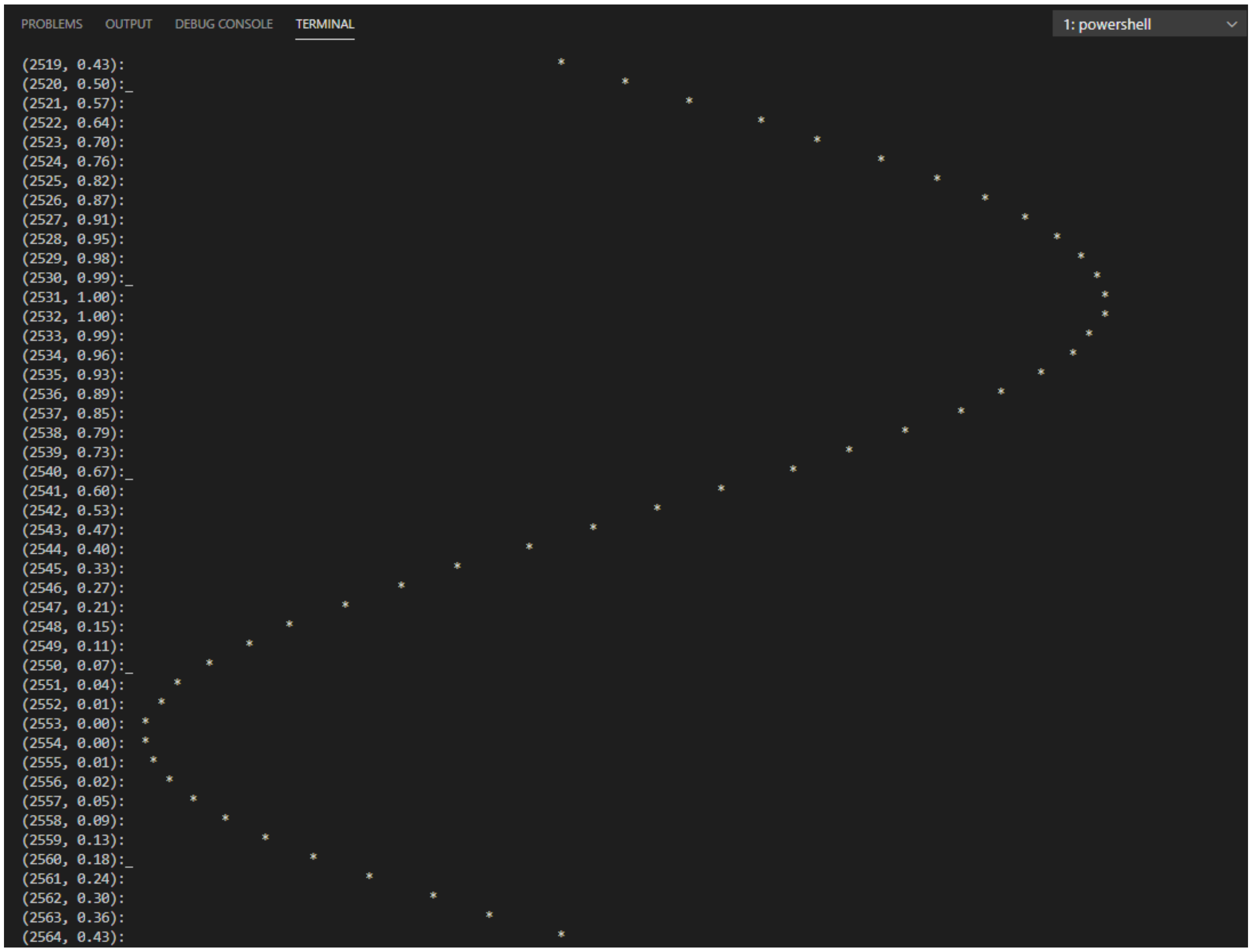
I then added a trace to the screen for scale. I added this every 10 lines like the provided example but this could easily be modified by changing the value in the if statement.



```c
/*Main code*/
int main()
{
    uint32_t x = 0;                             /*Declare x as fixed width unsigned variable, initialise equal to 0*/
    float y = 0;
    for (;;)                                    /*Endless loop*/
    {
        x++;                                    /*Increase x by 1*/
        printf("%8.0d, ", x);                   /*Print value of x to command line*/
        y = (sin(FREQ * x * pi / 180) + 1) / 2; /*calculate value of y*/
        printf("%0.2f:", y);                    /*Print value of y to command line*/
        if (x % 10 == 0)                        /*If multiple of 10 ad tick*/
        {
            printf("_");
        }
        else                                    /*Otherwise add space, so function is aligned*/
        {
            printf(" ");
        }
        plotval(y);                             /*Call Plotval*/
    }
    return SUCCESS;
}
```

```
    1273, 0.99:                                                                                                                         *
    1274, 0.96:                                                                                                                       *
    1275, 0.93:                                                                                                                   *
    1276, 0.89:                                                                                                              *
    1277, 0.85:                                                                                                         *
    1278, 0.79:                                                                                                  *
    1279, 0.73:                                                                                           *
    1280, 0.67:_                                                                                   *
    1281, 0.60:                                                                           *
    1282, 0.53:                                                                   *
    1283, 0.47:                                                           *
    1284, 0.40:                                                   *
    1285, 0.33:                                          *
    1286, 0.27:                                   *
    1287, 0.21:                            *
    1288, 0.15:                     *
    1289, 0.11:                *
    1290, 0.07:_          *
    1291, 0.04:       *
    1292, 0.01:     *
    1293, 0.00:   *
    1294, 0.00:   *
    1295, 0.01:    *
    1296, 0.02:      *
    1297, 0.05:         *
    1298, 0.09:             *
```

After this I changed the size of my fixed character width and added brackets so that it would neatly display the coordinates of the sine function.



```
(2519, 0.43):                                                       *
(2520, 0.50):_                                                              *
(2521, 0.57):                                                                       *
(2522, 0.64):                                                                                *
(2523, 0.70):                                                                                        *
(2524, 0.76):                                                                                                *
(2525, 0.82):                                                                                                       *
(2526, 0.87):                                                                                                             *
(2527, 0.91):                                                                                                                  *
(2528, 0.95):                                                                                                                      *
(2529, 0.98):                                                                                                                         *
(2530, 0.99):_                                                                                                                          *
(2531, 1.00):                                                                                                                            *
(2532, 1.00):                                                                                                                            *
(2533, 0.99):                                                                                                                          *
(2534, 0.96):                                                                                                                        *
(2535, 0.93):                                                                                                                    *
(2536, 0.89):                                                                                                               *
(2537, 0.85):                                                                                                          *
(2538, 0.79):                                                                                                   *
(2539, 0.73):                                                                                            *
(2540, 0.67):_                                                                                    *
(2541, 0.60):                                                                           *
(2542, 0.53):                                                                   *
(2543, 0.47):                                                           *
(2544, 0.40):                                                   *
(2545, 0.33):                                          *
(2546, 0.27):                                   *
(2547, 0.21):                            *
(2548, 0.15):                     *
(2549, 0.11):               *
(2550, 0.07):_         *
(2551, 0.04):      *
(2552, 0.01):    *
(2553, 0.00):  *
(2554, 0.00):  *
(2555, 0.01):   *
(2556, 0.02):     *
(2557, 0.05):        *
(2558, 0.09):            *
(2559, 0.13):                 *
(2560, 0.18):_                       *
(2561, 0.24):                               *
(2562, 0.30):                                       *
(2563, 0.36):                                              *
(2564, 0.43):                                                       *
```

The speed at which my program executes will be determined on the clock speed of the system running it. Visual Studio Code, my IDE, will have a clock speed that will do so many instructions a second. On embedded systems this will vary on the device. An arduino allows you to control the clock speed that you use to run the code through.