This is my transpose code including Computed includes so I just have to use this one transpose Code rather than making two separate c files.

```c
#if SYSTEM_1
# include "matrix_dynamic.h"
#elif SYSTEM_2
# include "matrix_static.h"
#endif

matrix transpose(matrix A_in) {
    int i,j;
    matrix A_out;
    A_out = create_empty(A_in.col_dim, A_in.row_dim);
    for(i=0; i<A_in.row_dim; i++) {
        for(j=0; j<A_in.col_dim; j++) {
            A_out.element[i][j] = A_in.element[j][i];
        }
    }
    return A_out;
}
```

Below is first the Dynamic code output and then the Static code outputs before I changed them to be integers

```
swyman2501@Knights-Castle:~/ECE330/matrixhw/Homework2$ ./Output_dynamic

Matrix a:

  1.000   2.000
  3.000   4.000

Matrix b:

  1.000   2.000
  3.000   4.000

a+b:

  2.000   4.000
  6.000   8.000

transposed a:

  1.000   3.000
  2.000   4.000

transposed b:

  1.000   3.000
  2.000   4.000
```

```
swyman2501@Knightys-Castle:~/ECE330/matrixhw/Homework2$ ./Output_static

Matrix a:

  1.000   2.000
  3.000   4.000

Matrix b:

  1.000   2.000
  3.000   4.000

a+b:

  2.000   4.000
  6.000   8.000

transposed a:

  1.000   3.000
  2.000   4.000

transposed b:

  1.000   3.000
  2.000   4.000
```

Now here is what I did to change the dataset into ints

```
 8 #define FORMAT2 "%8d"
 9 typedef int T;
10
```

I also changed the code in the print function to use FORMAT2 which I based off of FORMAT

Below is the outputs of the now changed Output_dynamic and Output_static codes

```
swyman2501@Knightys-Castle:~/ECE330/matrixhw/Homework2cLone$ ./Output_dynamic

Matrix a:

       1       2
       3       4

Matrix b:

       1       2
       3       4

a+b:

       2       4
       6       8

transposed a:

       1       3
       2       4

transposed b:

       1       3
       2       4
swyman2501@Knightys-Castle:~/ECE330/matrixhw/Homework2cLone$
```

```
swyman25010@Knightys-Castle:~/ECE350/matrixhw/Homework2clone$ ./Output_static

Matrix a:

      1       2
      3       4

Matrix b:

      1       2
      3       4

a+b:

      2       4
      6       8

transposed a:

      1       3
      2       4

transposed b:

      1       3
      2       4
```

Finally, below will be what I did for the Makefile

```
 1
 2 all: Output_static Output_dynamic
 3
 4 Output_static: matrix_static.c tandc.c test_static.c matrix_static.h
 5     gcc $^ -D SYSTEM_2 -o Output_static
 6
 7 Output_dynamic: matrix_dynamic.c tandc.c test_dynamic.c matrix_dynamic.h
 8     gcc $^ -D SYSTEM_1 -o Output_dynamic
 9
10 clean:
11     -rm Output_static
12     -rm Output_dynamic
```

This was actually my first time using computed includes so it was very interesting!