1. [20 pnts] Write a function that takes in a single integer and writes the table with that integer on the diagonal. You may assume the integer coming in is greater than or equal to 2. The program may use neither the while nor the do-while structure. (hint: it must use the for loop structure). Don’t just write a printf like printf("3 4 5 \n")).

The output, when a 3 is passed in, is:

| 3 4 5 |
| 2 3 4 |
| 1 2 3 |

The output, when a 5 is passed in, is:

| 5 6 7 8 9 |
| 4 5 6 7 8 |
| 3 4 5 6 7 |
| 2 3 4 5 6 |
| 1 2 3 4 5 |

#include <stdio.h>

void drawSquare(int i){
    int row, col;
    int startval = i;
    for (row = 0; row < i; row++){
        for (col = 0; col < i; col++){
            printf("%4d", startval+col);
        }
        printf("\n");
        startval--;
    }
    return;
}

int main(){
    drawSquare(3);
    drawSquare(5);
    return 0;
}
2. [10 pnts] Write the output of the following code. You do not need to worry about spaces - just make sure the correct values are on the correct lines.

```c
#include <stdio.h>

int f1(int i){
    int j = 5, k = 0;
    for (j = 10; j > i; j--){
        k+=i;
    }
    return k;
}

int f2(int i){
    int j, sum=i;
    for (j = 1; j < i; j++){  
        printf("%d: sum = %d\n",j,sum);
        sum +=i;
    }
    return sum;
}

int main(){
    printf("%d\n",f1(3));
    printf("%d\n",f2(5));
    return 0;
}
```

```
_21____________________________________________________________
_1: sum = 5____________________________________________________________
_2: sum = 10____________________________________________________________
_3: sum = 15____________________________________________________________
_4: sum = 20____________________________________________________________
_25____________________________________________________________
```