Amaan Asim

501086887

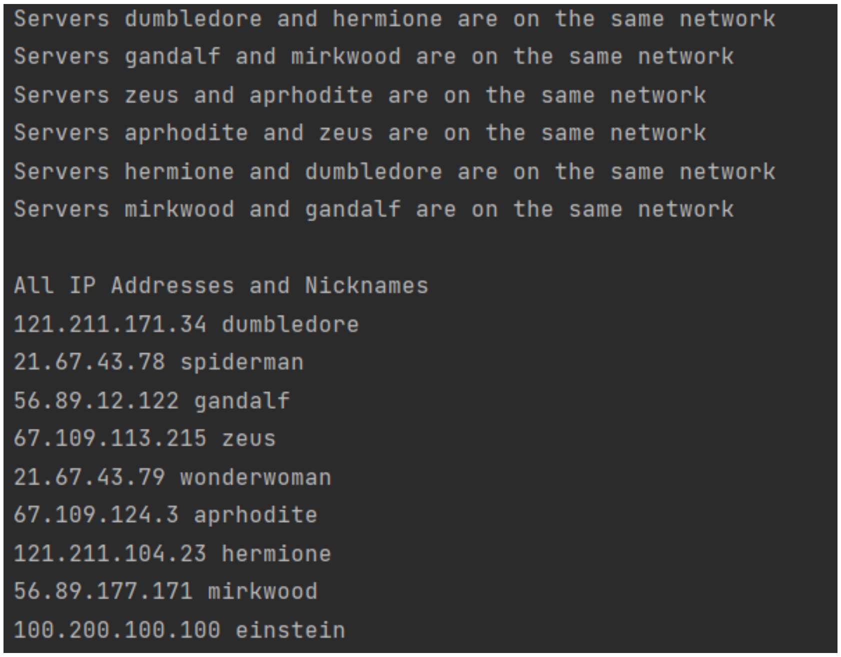
Section 07

Problem 1.

Code:

```
#include <stdio.h>
#include <string.h>
typedef struct address_t{
    int a;
    int b;
    int c;
    int d;
    char nick[16];
}address;
struct address_t e[300];

int localnet(address e1,address e2){
    int equal;
    if (e1.a == e2.a && e1.b == e2.b && e1.c != e2.c){
        equal = 1;
    }
    else{
        equal = 0;
    }
    return equal;
}

int main (void){
    int aa;
    int comp;
    int bb;
    int cc;
    int dd;
    char nickname[16];

    FILE *in;
    in = fopen("C:\\Users\\AmaanAsim\\Desktop\\ip.txt", "r");

    int i = 0;
    int j = 0;

    while((fscanf(in, "%d.%d.%d.%d %s", &aa, &bb, &cc, &dd, &nickname)!=EOF)){
        e[i].a = aa;
        e[i].b = bb;
        e[i].c = cc;
        e[i].d = dd;
        strncpy(e[i].nick, nickname, 16);

        i++;
    }
    i = 0;
    for (i = 0; i<11; ++i){
        for(j = 0; i<11; ++j){
            comp = localnet(e[i], e[j]);
            if (comp == 1){
                printf("Servers %s and %s are on the same network\n", e[i].nick, e[j].nick);
            }
        }
    }

    printf("\nAll IP Addresses and Nicknames\n");
    for (i=0; i<11; ++i){
        if(e[i].a != 0){
            printf("%d.%d.%d.%d %s\n", e[i].a, e[i].b, e[i].c, e[i].d, e[i].nick) ;
        }
        else{
            fclose(in);
        }
    }
}
```

Output:



```
Servers dumbledore and hermione are on the same network
Servers gandalf and mirkwood are on the same network
Servers zeus and aprhodite are on the same network
Servers aprhodite and zeus are on the same network
Servers hermione and dumbledore are on the same network
Servers mirkwood and gandalf are on the same network

All IP Addresses and Nicknames
121.211.171.34 dumbledore
21.67.43.78 spiderman
56.89.12.122 gandalf
67.109.113.215 zeus
21.67.43.79 wonderwoman
67.109.124.3 aprhodite
121.211.104.23 hermione
56.89.177.171 mirkwood
100.200.100.100 einstein
```

Problem 2.

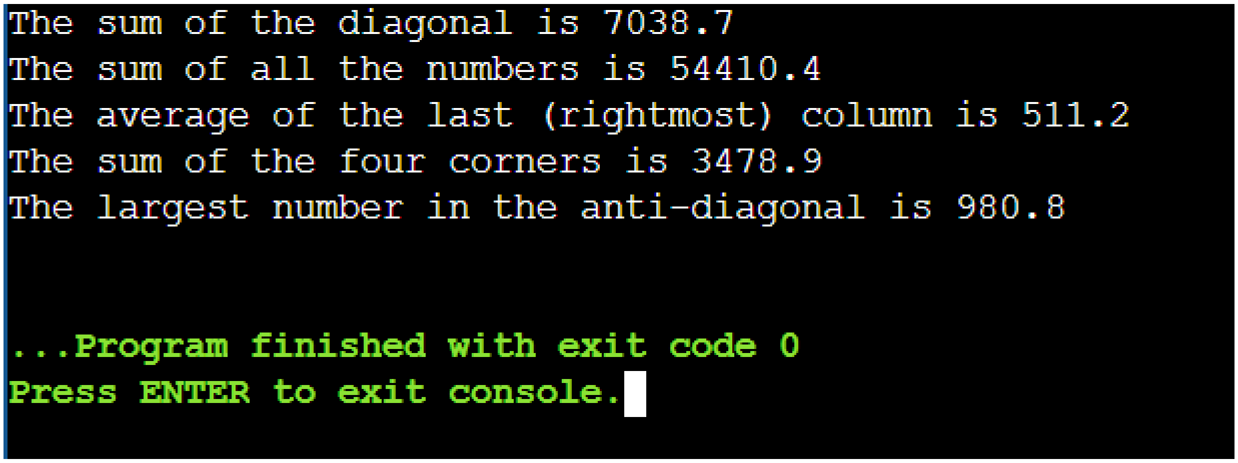
Code:

```
#include <stdio.h>
#include "mylibrary.h"
struct answers{
    int n1, n2, n3, n4, n5;
};

int main(void){
    double matrix[10][10];
    int i, j;
    FILE *in;
    in=fopen("array.txt", "r");
    for (i=0; i<10; ++i)
    for (j=0; j<10; ++j)
    fscanf (in, "%lf", &matrix[i][j]);

    fclose(in);

    //for (i=0; i<10; ++i)
    //{
   //     for (j=0; j<10; ++j)
  //      printf ("%.2lf ", matrix[i][j]);
   //     printf ("\n");
    //}

    double diagsum;
    diagsum = sumdiag(10, matrix);
    //printf("\n%.2lf", diagsum);

    double allsum;
    allsum = sumall(10, matrix);
   // printf("\n%.2lf", allsum);

    double rightav;
    rightav = avright(matrix);
   // printf("\n%.2lf", rightav);

    double cornersum;
    cornersum = corners(matrix);
    //printf("\n%.2lf", cornersum);

    double largestanti;
    largestanti = largeanti(matrix);
   // printf("\n%.2lf\n", largestanti);


    FILE *bin;
    bin = fopen("results.bin", "wb");

    double results[5] = {diagsum, allsum, rightav, cornersum, largestanti};

    fwrite(results, sizeof (double), 5, bin);
    fclose(bin);

    bin = fopen("results.bin", "rb");

    double answers[5];
    fread(answers, sizeof (double), 5, bin);
    printf("The sum of the diagonal is %.1lf\n", answers[0]);
    printf("The sum of all the numbers is %.1lf\n", answers[1]);
    printf("The average of the last (rightmost) column is %.1lf\n", answers[2]);
    printf("The sum of the four corners is %.1lf\n", answers[3]);
    printf("The largest number in the anti-diagonal is %.1lf\n", answers[4]);

}
```

Library:

```
#ifndef UNTITLED_MYLIBRARY_H
#define UNTITLED_MYLIBRARY_H

#endif //UNTITLED_MYLIBRARY_H

double sumdiag(int n, double array[n][n]){
    double sum = 0;
    for(int i = 0; i < n; i++){
        sum += array[i][i];
    }
    return sum;
}

double sumall(int n, double array[n][n]){
    double sum = 0;
    for(int i = 0; i < 10; ++i){
        for(int j = 0; j < 10; ++j){
            sum += array[i][j];
        }
    }
    return sum;
}

double avright(double array[10][10]){
    double sum = 0;
    double average;
    int count = 0;
    for(int i = 0; i < 10; ++i){
        sum += array[i][9];
        count++;
    }
    average = sum/count;
    return average;
}

double corners(double array[10][10]){
    double sum = 0;
    sum = array[0][0] + array [0][9] + array [9][0] + array [9][9];
    return sum;
}

double largeanti(double array[10][10]) {
    int j = 9;
    int largest = 0;
    for (int i=0; i<10; i++){
        if (array[i][j]>largest) {
            largest = array[i][j];
        }
        j --;
    }
    return largest;
}
```

Output:



```
The sum of the diagonal is 7038.7
The sum of all the numbers is 54410.4
The average of the last (rightmost) column is 511.2
The sum of the four corners is 3478.9
The largest number in the anti-diagonal is 980.8


...Program finished with exit code 0
Press ENTER to exit console.
```