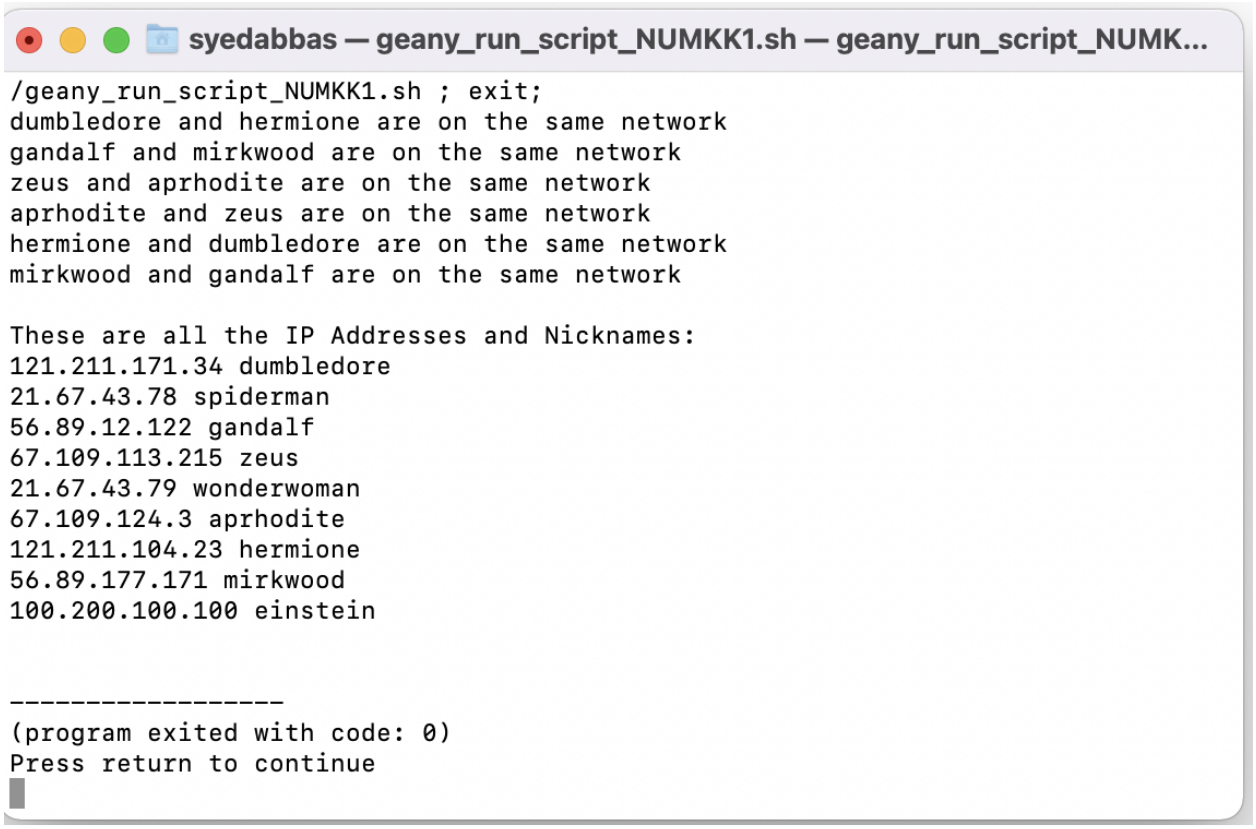
Structures, Binary files, and Personal Libraries

Question 1

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

typedef struct address_t
{
	int a, b, c, d;
	char fifth_comp[16];
}
address;
struct address_t file[300];

int
localnet (address file1, address file2)
{
	int network;
	if (file1.a == file2.a && file1.b == file2.b && file1.c != file2.c)
	{
		network = 1;
	}

		else
		{
			network = 0;
		}

return network;
}

int
main (void)
{
	int aa, bb, cc, dd, temp;
	char component_5[16];
```

```
FILE *in;
in = fopen("sentinel_file.txt", "r");

int i=0, j=0;

while ((fscanf(in, "%d.%d.%d.%d %s", &aa, &bb, &cc, &dd, &component_5)!=EOF))
{
	file[i].a = aa;
	file[i].b = bb;
	file[i].c = cc;
	file[i].d = dd;
	strncpy(file[i].fifth_comp, component_5, 16);
	i++;
}

i = 0;

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

printf("\nThese are all the IP Addresses and Nicknames:\n");

for (i=0; i<11; ++i)
{
	if(file[i].a != 0)
	{
printf("%d.%d.%d.%d %s\n", file[i].a, file[i].b, file[i].c, file[i].d, file[i].fifth_comp) ;
	}
		else
		{
		fclose(in);
```

```
        }
}
}
```



Question 2

```
#include <stdio.h>
#include "mylibrary.h"

struct results
{
        int ans1, ans2, ans3, ans4, ans5;
};

int
main (void)
{

double matrix[10][10];
int i, j;
```

```
FILE *in;
in=fopen("array_file.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 sum_of_diag;
sum_of_diag = sumdiag(10, matrix);
printf ("\n%.2lf", sum_of_diag);

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

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

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

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

FILE *bin;
```

```
bin = fopen("results.bin", "wb");

double ans[5] = {sum_of_diag, total_sum, average_right, sum_corners, large_antidiag};
fwrite(ans, sizeof (double), 5, bin);
fclose(bin);

bin = fopen("results.bin", "rb");
double results[5];

fread(results, sizeof (double), 5, bin);
printf("\nThe sum of the main diagonal is %.1lf\n", results[0]);
printf("The total sum of all the elements is %.1lf\n", results[1]);
printf("The average of the last (rightmost) column is %.1lf\n", results[2]);
printf("The sum of the four corners is %.1lf\n", results[3]);
printf("The largest number in the anti-diagonal is %.1lf\n", results[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, 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, largest = 0;
        for (int i=0; i<10; i++)
        {
                if (array[i][j]>largest)
                {
                largest = array[i][j];
                }
        --j;
        }

return largest;
}
```

```
The sum of the main diagonal is 7038.7
The total sum of all the elements 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 980.0


------------------
(program exited with code: 0)
Press return to continue
```