```c
//Name: Mehmet Fatih Celik
//ID: 2385268

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

struct RGB_Image{
    int R[5][5];
    int G[5][5];
    int B[5][5];
};

struct grayScale_Image{
    int myArray[5][5];
};

struct RGB_Image *formRGBImage(int num){
    int i,j,k;
    srand(time(NULL));
    struct RGB_Image *my_image;
    my_image= (struct RGB_Image*)malloc(num*sizeof(struct RGB_Image));

    for(k=0;k<num;k++){
        for(i=0;i<5;i++){
            for(j=0;j<5;j++){
                my_image[k].R[i][j] = rand() %256;
                my_image[k].G[i][j] = rand() %256;
                my_image[k].B[i][j] = rand() %256;
            }
        }
    }
    return my_image;
}

struct grayScale_Image *thresholding(struct RGB_Image *my_image, int thresholding_num, int num){
    int i,j,k;
    struct grayScale_Image *grayScale;
    grayScale = (struct grayScale_Image*)malloc(num*sizeof(struct grayScale_Image));
    for(k=0;k<num;k++){
        for(i=0;i<5;i++){
            for(j=0;j<5;j++){
                if (thresholding_num > (my_image[k].R[i][j]+ my_image[k].G[i][j]
+my_image[k].B[i][j])/3)
                    grayScale[k].myArray[i][j] = 0;
                else
                    grayScale[k].myArray[i][j] = 1;
            }
        }
    }
    return grayScale;
```

```
}

void displayThresholdedImage(struct grayScale_Image *grayScale, int num){
    int i,j,k;
    for(k=0;k<num;k++){
        printf("\n******Image %d*****\n",k+1);
        for(i=0;i<5;i++){
            for(j=0;j<5;j++){
                if (j==4){
                    printf("%d\n",grayScale[k].myArray[i][j]);
                    continue;
                }
                printf("%d\t",grayScale[k].myArray[i][j]);
            }
        }
    }
}

int main(){
    struct RGB_Image *my_image;
    struct grayScale_Image *grayScale;
    int num, thresholding_num;
    printf("Please enter how many images you want to create: ");
    scanf("%d",&num);
    my_image = formRGBImage(num);
```

```
        printf("Please enter the thresholding value: ");
        scanf("%d",&thresholding_num);


        grayScale = thresholding(my_image, thresholding_num, num);


        displayThresholdedImage(grayScale, num);


        return 0;
}
```