1) In the following table, a part of bitmap image of 8x8 is given. The number of colors in the image is 8 bit per pixel (the gray levels are 0-255). Apply the following high pass filter to the part of image and show the new gray values (25 Points)

-1	-1	2
-1	2	-1
2	-1	-1

119	118	212	215	235	8	144	137
94	84	1	125	244	123	17	255
153	73	173	87	70	103	42	157
127	210	225	207	33	139	29	33
90	145	56	148	249	12	0	202
232	203	95	25	246	240	60	222
218	234	156	20	117	164	13	37
88	252	68	255	134	119	5	107

Solution:

119	118	212	215	235	8	144	137
94	255	0	237	0	25	0	255
153	0	255	255	69	0	255	157
127	167	47	0	0	255	0	33
90	255	84	0	0	0	0	202
232	2	255	178	0	0	255	222
218	0	0	0	255	0	95	37
88	252	68	255	134	119	5	107

2) Calculate the total bitmap file size of the following image type, heigth and width given.(25 Points)

Per pixel size	Height	Width	File Header	Info Header	Color Palette	Data	Total Size
1	100	25	14	40	8	400	462
4	100	89	14	40	64	4800	4918
8	100	121	14	40	1024	12400	13478
24	100	99	14	40	-	30000	30054

3) Complete the following programming function that is used to strecth an bitmap image. Please add BYTE findMin(IMAGE *image) and BYTE findMax(IMAGE*image) functions. The streching formula is: (25 points)

$$g(r,c) = \frac{(MAX - MIN)(l(r,c) - l(r,c)_{min})}{l(r,c)_{max} - l(r,c)_{min}} + MIN$$

$Where$:
$l(r,c)$: $original\ pixel\ value\ of\ image$
$g(r,c)$: $the\ new\ pixel\ value\ of\ image\ after\ strecthing$

$l(r,c)_{min}$: $minimum\ pixel\ value\ of\ bitmap\ image$, $l(r,c)_{max}$: $maximum\ pixel\ value\ of\ bitmap\ image$
MAX: $after\ strecthing\ the\ maximum\ pixel\ value\ of\ value\ of\ image$, $\max\{g(r,c)\}$
MIN: $after\ strecthing\ the\ minimum\ pixel\ value\ of\ image$,, $\min\{g(r,c)\}$

```
BYTE findMin(IMAGE *image)
{
        int i,j,h,w,rowsize;
        h=image->bmpih.bih;
        w=image->bmpih.biw;
        rowsize=(image->bmpih.biw*image->bmpih.bibitcount+31)/32*4;
        BYTE min=image->data[0];
        for(i=0;i<h*rowsize;i++) if(image->data[i]<min) min=image->data[i];
        return min;
}
BYTE findMax(IMAGE *image)
{
        int i,j,h,w,rowsize;
        h=image->bmpih.bih;
        w=image->bmpih.biw;
        rowsize=(image->bmpih.biw*image->bmpih.bibitcount+31)/32*4;
        BYTE max=image->data[0];
        for(i=0;i<h*rowsize;i++) if(image->data[i]>max) max=image->data[i];
        return max;
}
void F(IMAGE *image, BYTE MIN,BYTE MAX)
{
        int i,j,h,w,rowsize;
        h=image->bmpih.bih;
        w=image->bmpih.biw;
        rowsize=(image->bmpih.biw*image->bmpih.bibitcount+31)/32*4;
        BYTE min=findMin(image);
        BYTE max=findMax(image);
        for(i=0;i<h*rowsize;i++)
           image->data[i]=(MAX-MIN)*image->data[i]-min)/(max-min)+min;
}
```

4) The following function we want to apply higpass filter mask of 3x3 given in the q1. Please complete the follwing function. (25 points)

```
void F(IMAGE *image,int mask[3][3])
{
        int h,w,rowsize,i,j,k,n,m,l;
        BYTE *data;
        int sum;
        w=image->bmpih.biw;
        h=image->bmpih.bih;
        rowsize=(image->bmpih.bibitcount*w+31)/32*4;
        data=(BYTE*)malloc(h*rowsize*sizeof(BYTE));
  memcpy(data,image->data,h*rowsize);
        for(i=1;i<h-1;i++)
                for(j=1;j<rowsize-1;j++)
                {
                        sum=0;
                        for(n=-1;n<=1;n++)
                          for(m=-1;m<=1;m++)
                            sum+=Filter[n+1][m+1]*data[(i+n)*rowsize+j+m];
                        if(sum<0)sum=0;
                        else if(sum>255)sum=255;
                        image->data[i*rowsize+j]=sum;
        }
  free(data);
  return;

}
```