NAME: AMISHA USN: 1RV17CS017 COURSE: PADP (16CS71)

program 4 : Write a Program to convert color img to Black & white image

```
#include<stdio.h>
#include<mach/error.h>
#include<gd.h>
#include<string.h>
#include<omp.h>
int main(int argc, char **argv){
    FILE *fp, *fp1 = {0};
    gdImagePtr img;
    char iname[15], oname[15];
    int color, x, y, i=0, red, green, blue, tmp;
    long w, h;
    omp_sched_t def_sched;
    int def_chunk_size;
    omp_get_schedule(&def_sched, &def_chunk_size);
    printf("Default %d %d\n", def_sched, def_chunk_size);
    printf("Size\t\tDefault\t\tstatic\t\tDynamic\t\t
    guided\n");
    for(int i=1; i<=5; i++){
      sprintf(iname, "in%d.png", i);
      for(int sched=0x0; sched<=0x3; sched++){
        fp = fopen(iname, "r");
        sprintf(oname, "Output%d%d.png", i, sched);
        img = gdImageCreateFromPng(fp);
        w = gdImageSX(img);
        h = gdImageSY(img);
        if(sched==0x0){
          printf("%ldx%ld\t\t", w, h);
          omp_set_schedule(def_sched, def_chunk_size);
        }
      else
        omp_set_schedule(sched, 0);
      double t = omp_get_wtime();
      #pragma omp parallel for private(y, color,
              red, green, blue, tmp)
```

```
for (x=0; x<w; x++)
    for (y=0; y<h; y++) {
        color = gdImageGetPixel (img, x, y);
        red = gdImageRed (img, color);
        green = gdImageGreen (img, color);
        blue = gdImageBlue (img, color);
        tmp = (red + green + blue) / 3;
        red = green = blue = tmp;
        color = gdImageColorAllocate (img, red, green
        blue);
        gdImageSetPixel (img, x, y, color);
        }
    }
    t = omp_get_wtime() - t;
    fp1 = fopen (oname, "w");
    gdImagePng (imp, fp1);
    fclose (fp1);
    gdImageDestroy (imp);
    printf ("%.6f\t", t);
    }
    printf ("\n");
    }
return 0;
}
```