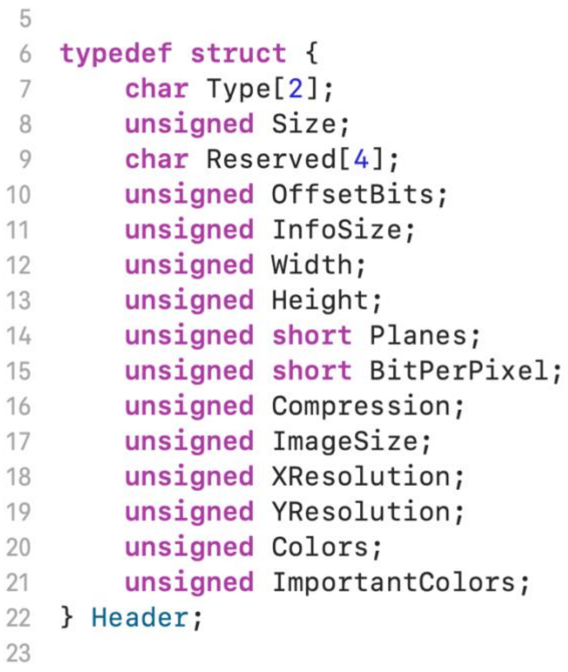
Programming Assignment 3:

Image Reduction and Merge with Frame

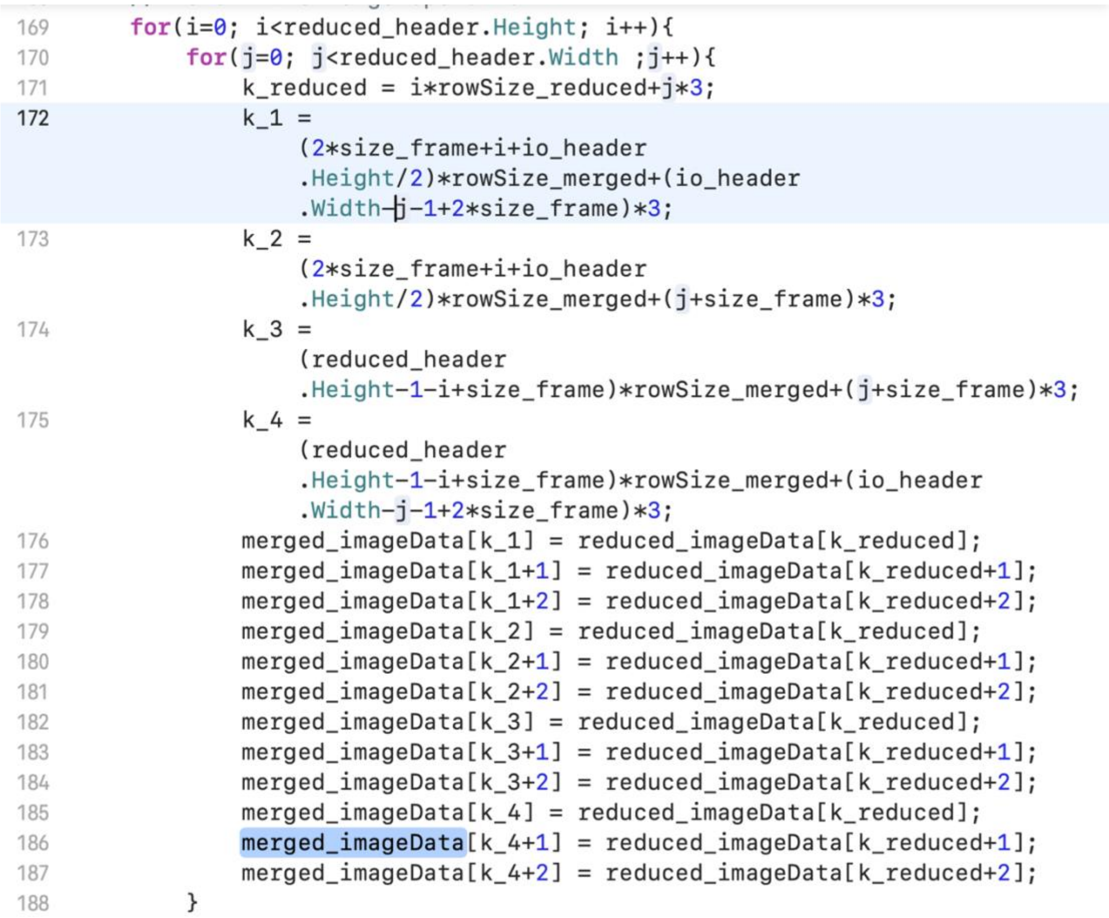
Name: Derek (劉哲瑋) Number: D1262032

I first define a structure named Header, as shown in the picture below.



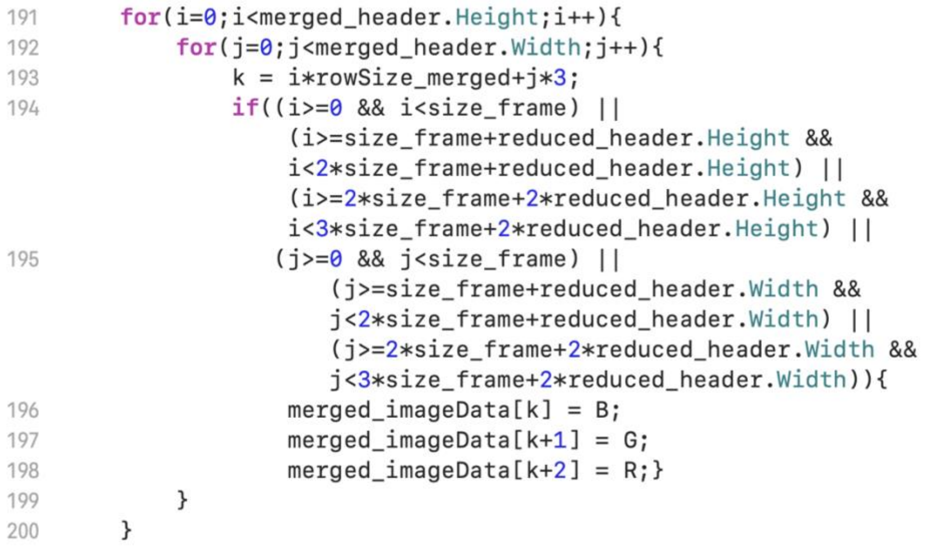
```c

typedef struct {
    char Type[2];
    unsigned Size;
    char Reserved[4];
    unsigned OffsetBits;
    unsigned InfoSize;
    unsigned Width;
    unsigned Height;
    unsigned short Planes;
    unsigned short BitPerPixel;
    unsigned Compression;
    unsigned ImageSize;
    unsigned XResolution;
    unsigned YResolution;
    unsigned Colors;
    unsigned ImportantColors;
} Header;

```

Then, I use “fread” to input the colored image file. Next, I use “malloc” to allocate memory space for the palette and image pixel data. And I use “fread” to read palette from the image file and to read image pixel data from the image file. Next, I use the functions I defined, which are named “print_header” to print the header and “write_image_file” to write the image file. Then I write the image bitmap file of the reduced image to disk and output its file header and the image information head on the screen. Then I create a header called merged_header. Next, I use the code below to perform the merge operation.



```
    for(i=0; i<reduced_header.Height; i++){
        for(j=0; j<reduced_header.Width ;j++){
            k_reduced = i*rowSize_reduced+j*3;
            k_1 =
                (2*size_frame+i+io_header
                .Height/2)*rowSize_merged+(io_header
                .Width-j-1+2*size_frame)*3;
            k_2 =
                (2*size_frame+i+io_header
                .Height/2)*rowSize_merged+(j+size_frame)*3;
            k_3 =
                (reduced_header
                .Height-1-i+size_frame)*rowSize_merged+(j+size_frame)*3;
            k_4 =
                (reduced_header
                .Height-1-i+size_frame)*rowSize_merged+(io_header
                .Width-j-1+2*size_frame)*3;
            merged_imageData[k_1] = reduced_imageData[k_reduced];
            merged_imageData[k_1+1] = reduced_imageData[k_reduced+1];
            merged_imageData[k_1+2] = reduced_imageData[k_reduced+2];
            merged_imageData[k_2] = reduced_imageData[k_reduced];
            merged_imageData[k_2+1] = reduced_imageData[k_reduced+1];
            merged_imageData[k_2+2] = reduced_imageData[k_reduced+2];
            merged_imageData[k_3] = reduced_imageData[k_reduced];
            merged_imageData[k_3+1] = reduced_imageData[k_reduced+1];
            merged_imageData[k_3+2] = reduced_imageData[k_reduced+2];
            merged_imageData[k_4] = reduced_imageData[k_reduced];
            merged_imageData[k_4+1] = reduced_imageData[k_reduced+1];
            merged_imageData[k_4+2] = reduced_imageData[k_reduced+2];
        }
```

Then, I use 2 for loops to add the framed area, as shown below.



```
for(i=0;i<merged_header.Height;i++){
    for(j=0;j<merged_header.Width;j++){
        k = i*rowSize_merged+j*3;
        if((i>=0 && i<size_frame) ||
            (i>=size_frame+reduced_header.Height &&
            i<2*size_frame+reduced_header.Height) ||
            (i>=2*size_frame+2*reduced_header.Height &&
            i<3*size_frame+2*reduced_header.Height) ||
           (j>=0 && j<size_frame) ||
               (j>=size_frame+reduced_header.Width &&
               j<2*size_frame+reduced_header.Width) ||
               (j>=2*size_frame+2*reduced_header.Width &&
               j<3*size_frame+2*reduced_header.Width)){
            merged_imageData[k] = B;
            merged_imageData[k+1] = G;
            merged_imageData[k+2] = R;}
    }
}
```

Finally, I use “free()” to release memory space of “io_palette”, “io_imageData”, “reduced_imageData” and “merged_imageData”.