```cpp
/** \author MATTHIAS_KOCKISCH
 *  \date 10 dec 2018
 */
#include <iostream>
#include <vector>
#include <array>
#include <cmath>
using namespace std;

//TYPEDEF
typedef vector<vector<array<int,3> > > Image;
typedef vector<vector<int> > ImageS;

//INPUT
int cin_nbR();
vector<array<int,3> > cin_cf(int nbR);
vector<double> cin_seuil(int nbR);
int cin_nbF();
Image cin_image(int nbL,int nbC,int MAX);

//SEUILLAGE
ImageS seuil_images(int nbL,int nbC,int MAX,int nbR,Image image0,vector<double> seuil);

//FILTRAGE
ImageS prefiltre(int nbF,int nbL,int nbC,ImageS images0, ImageS images1);
int filtre(array<int,8> tf);
ImageS bords_noirs(ImageS images0, int nbL, int nbC);

//SORTIES
void cout_RVB(int nbL, int nbC, ImageS images0, vector<array<int,3> > cf);

//ERROR
void error_nbR(int nbR);
void error_color(int id);
void error_threshold(double invalid_val);
void error_nb_filter(int nb_filter);

//MAIN
int main() {

  //INPUT
  int nbR(cin_nbR());
  vector<array<int,3> > cf(nbR+1);
  cf=cin_cf(nbR);
  vector<double> seuil(nbR+1);
  seuil=cin_seuil(nbR);
  int nbF(cin_nbF());
  char P3[2];
  int nbC, nbL, MAX;
  cin >> P3 >> nbC >> nbL >> MAX;
  Image image0(nbL, vector<array<int,3> >(nbC));
  image0=cin_image(nbL, nbC, MAX);

```

```
54    //SEUILLAGE
55    ImageS images0(nbL, vector<int>(nbC));
56    images0=seuil_images(nbL, nbC, MAX, nbR, image0, seuil);
57
58    //FILTRAGE
59    ImageS images1(nbL, vector<int>(nbC));
60    images0=prefiltre(nbF, nbL, nbC, images0, images1);
61    if (nbF>0) images0=bords_noirs(images0, nbL, nbC);
62
63    //SORTIES
64    cout << P3 << endl << nbC << " " << nbL << endl << MAX << endl;
65    cout_RVB(nbL, nbC, images0, cf);
66    return 0;
67  }
68
69  int cin_nbR() {
70    int nbR;
71    cin >> nbR;
72    if (nbR<2 or nbR>255) {
73      error_nbR(nbR);
74      exit(0);
75    }
76    return nbR;
77  }
78
79  vector<array<int,3> > cin_cf(int nbR) {
80    vector<array<int,3> > cf(nbR+1);
81    for (int i(0); i < nbR; i++) {
82      for (int j(0); j < 3; j++) {
83        cf[0][j]=0;
84        cin >> cf[i+1][j];
85        if (cf[i+1][j]<0 or cf[i+1][j]>255) {
86          error_color(i+1);
87          exit(0);
88        }
89      }
90    }
91    return cf;
92  }
93
94  vector<double> cin_seuil(int nbR) {
95    vector<double> seuil(nbR+1);
96    seuil[0]=0;
97    seuil[nbR]=1;
98    for (int i(1); i<nbR; i++) {
99      cin >> seuil[i];
100     if (seuil[i]-seuil[i-1]<0.001 or seuil[i]>1 or seuil[i]<0) {
101       error_threshold(seuil[i]);
102       exit(0);
103     }
104   }
105   return seuil;
106 }
```

```
107
108  int cin_nbF() {
109    int nbF;
110    cin >> nbF;
111    if (nbF<0) {
112      error_nb_filter(nbF);
113      exit(0);
114    }
115    return nbF;
116  }
117
118  Image cin_image(int nbL,int nbC,int MAX) {
119    Image image0(nbL, vector<array<int,3> >(nbC));
120    for (int i(0); i<nbL; i++) {
121      for (int j(0); j<nbC; j++) {
122        for (int k(0); k<3; k++) {
123          cin >> image0[i][j][k];
124          if (image0[i][j][k]<0 or image0[i][j][k]>MAX) {
125            error_color(image0[i][j][k]);
126            exit(0);
127          }
128        }
129      }
130    }
131    return image0;
132  }
133
134  //SEUILLAGE
135  ImageS seuil_images(int nbL,int nbC,int MAX,int nbR,Image image0,vector<double> seuil){
136    ImageS images0(nbL, vector<int>(nbC));
137    for (int i(0); i<nbL; i++) {
138      for (int j(0); j<nbC; j++) {
139        int c(0), s(0);
140        double in(0);
141        for (int k(0); k<3; k++) {
142          c += pow(image0[i][j][k],2);
143        }
144        in =sqrt(c/(3*pow(MAX,2)));
145        while (in >= seuil[s] and s<nbR) {
146          s++;
147        }
148        images0[i][j] = s;
149      }
150    }
151    return images0;
152  }
153
154
155
156
157
158
159
```

```
160  //FILTRAGE
161  ImageS prefiltre (int nbF,int nbL,int nbC,ImageS images0, ImageS images1){
162    images1=images0;
163    for (int f(0); f<nbF; f++) {
164      for (int i(1); i<nbL-1; i++) {
165        for (int j(1); j<nbC-1; j++) {
166          array<int,8> tf {images0[i-1][j-1],images0[i-1][j],
167            images0[i-1][j+1],images0[i][j-1],images0[i][j+1],
168            images0[i+1][j-1],images0[i+1][j],images0[i+1][j+1]};
169          images1[i][j]=filtre(tf);
170        }
171      }
172
173      images0=images1;
174    }
175    return images0;
176  }
177
178  int filtre(array<int,8> tf) {
179    int c(0), d(0);
180    do {
181      d=0;
182      do {
183        c++;
184        } while (tf[c]==tf[c+1]);
185      for (int e(0); e<8; e++) {
186        if (tf[c]==tf[e]) {
187          d++;
188        }
189      }
190    } while (c<3 and d<6);
191    if (d>5) return tf[c];
192    else return 0;
193  }
194
195  ImageS bords_noirs(ImageS images0, int nbL, int nbC) {
196    for (int i(0); i<nbL; i++) {
197      images0[i][0]=0;
198      images0[i][nbC-1]=0;
199    }
200    for (int j(0); j<nbC; j++) {
201      images0[0][j]=0;
202      images0[nbL-1][j]=0;
203    }
204    return images0;
205  }
206
207
208
209
210
211
212
```

```
213  //SORTIES
214  void cout_RVB(int nbL, int nbC, ImageS images0, vector<array<int,3> > cf) {
215    for (int i(0); i<nbL; i++) {
216      for (int j(0); j<nbC; j++) {
217        for (int k(0); k<3; k++) {
218          cout << cf[images0[i][j]][k] << " ";
219        }
220      }
221      cout << endl;
222    }
223    cout << endl;
224  }
225
226  //ERROR
227  void error_nbR(int nbR)
228  {
229    cout << "Invalid number of colors: " << nbR << endl;
230  }
231  void error_color(int id)
232  {
233    cout << "Invalid color value " << id << endl;
234  }
235  void error_threshold(double invalid_val)
236  {
237    cout << "Invalid threshold value: " << invalid_val << endl;
238  }
239  void error_nb_filter(int nb_filter)
240  {
241    cout << "Invalid number of filter: " << nb_filter << endl;
242  }
243
```