

## 最終課題B：1次元バーコードの読み取りプログラムおよび解説とその実行結果レポート

---

1次元バーコードの読み取りプログラムおよび解説とその実行結果

### 目次

コード	2
解説及び実行結果	7

---

## コード

```
#include <stdio.h>
#include "pgmlib.h"
#include <math.h>

// しきい値での二値化
void binarize_image(int index,int threshold) {
    int x, y, val;
    for (y = 0; y < height[index]; y++) {
        for (x = 0; x < width[index]; x++) {
            val = image[index][x][y];

            if (val < threshold) {
                image[index][x][y] = 0;
            } else {
                image[index][x][y] = 255;
            }
        }
    }
}

// バーコード方向への黒（0）と白（255）の値の連続個数を数えて順に表示する
void countConsecutiveValues(int *inputArray, int length, int *output, int
outputLength) {
    int count = 1;
    int currentValue = inputArray[0];
    int outputIndex = 0;

    for (int i = 1; i < length; i++) {
        if (inputArray[i] == currentValue) {
            count++;
        } else {
            // printf("連続する %d の数: %d\n", currentValue, count);
            int num;
            num = round(count / 8.5); //最初の連続した黒と白のピクセル数で適当に割る
            if (outputIndex < outputLength) {
                output[outputIndex] = num;
                outputIndex++;
            } else {
                printf("出力配列のサイズが足りません\n");
                return;
            }

            currentValue = inputArray[i];
            count = 1;
        }
    }
}
```

```

}

int main(void){

    // しきい値 125 での二値化
    load_image(0, "バーコード 9784254122060.pgm");
    copy_image(0,1);
    binarize_image(1, 125);
    save_image(1, "バーコード 9784254122060_125.pgm");

    // バーコードの 50 行目の階調値を配列 Array に保存
    load_image(2, "バーコード 9784254122060_125.pgm");
    int Array[width[2]];
    int val,x,y=50;
    for(x=0;x<width[2];x++){
        val = image[2][x][y];
        Array[x] = val;
    }

    // Array 配列の内容を出力
    printf("¥nArray 配列の内容¥n");
    for (int i = 0; i < sizeof(Array) / sizeof(Array[0]); i++) {
        printf("%d ", Array[i]);
    }
    printf("¥n¥n");

    // 配列の長さを計算
    int arrayLength = sizeof(Array) / sizeof(Array[0]);
    int output[61];
    countConsecutiveValues(Array, sizeof(Array) / sizeof(Array[0]), output,
sizeof(output) / sizeof(output[0]));

    // output 配列の内容を出力
    printf("output 配列の内容");
    for (int i = 0; i < sizeof(output) / sizeof(output[0]); i++) {
        printf("%d ", output[i]);
    }
    printf("¥n¥n");
}

```

```

// JAN コード→バーコード 配列作る
int JANcode_Barcode[10][3][4] = {
{{3,2,1,1},{1,1,2,3},{3,2,1,1}},
{{2,2,2,1},{1,2,2,2},{2,2,2,1}},
{{2,1,2,2},{2,2,1,2},{2,1,2,2}},
{{1,4,1,1},{1,1,4,1},{1,4,1,1}},
{{1,1,3,2},{2,3,1,1},{1,1,3,2}},
{{1,2,3,1},{1,3,2,1},{1,2,3,1}},
{{1,1,1,4},{4,1,1,1},{1,1,1,4}},
{{1,3,1,2},{2,1,3,1},{1,3,1,2}},
{{1,2,1,3},{3,1,2,1},{1,2,1,3}},
{{3,1,1,2},{2,1,1,3},{3,1,1,2}}
};

int array_[13]; // 1~12 は識別してきた jan コードを保存、0 は最後の国番号を保存
int country[6]; // 識別してきた奇数 0・偶数 1 を保存、奇数偶数の並びで国番号を識別する

int count_array = 0; // 配列に記入する index
for(int n=4; n<27;n+=4){
    int target[4] = {output[n],output[n+1],output[n+2],output[n+3]}; // 識別したい
4 桁
    int found = 0;

    for (int i = 0; i < 10; ++i) {
        for (int j = 0; j < 3; ++j) {
            int match = 1;
            for (int k = 0; k < 4; ++k) {
                if (JANcode_Barcode[i][j][k] != target[k]) {
                    match = 0;
                    break;
                }
            }
            if (match) {
                printf("[%d,%d,%d,%d] の JAN コードは[%d]行[%d]列にあります¥n",
output[n],output[n+1],output[n+2],output[n+3],i,j);
                array_[count_array+1] = i;
                country[count_array] = j;
                count_array+=1;

                found = 1;
                break;
            }
        }
        if (found) break;
    }

    if (!found) {

```

```

        printf("%d,%d,%d,%d は見つかりませんでした。
¥n",output[n],output[n+1],output[n+2],output[n+3]);
    }
}

for(int n=33; n<54;n+=4){
    int target[4] = {output[n],output[n+1],output[n+2],output[n+3]};
    int found = 0;

    for (int i = 0; i < 10; ++i) {
        for (int j = 0; j < 3; ++j) {
            int match = 1;
            for (int k = 0; k < 4; ++k) {
                if (JANcode_Barcode[i][j][k] != target[k]) {
                    match = 0;
                    break;
                }
            }
            if (match) {
                printf("[%d,%d,%d,%d]の JAN コードは[%d]行にあります¥n",
output[n],output[n+1],output[n+2],output[n+3],i);
                array_[count_array+1] = i;
                count_array+=1;

                found = 1;
                break;
            }
        }
        if (found) break;
    }

    if (!found) {
        printf("1,3,1,2 は見つかりませんでした。 ¥n");
    }
}

// 国番号配列の内容を出力
printf("国番号配列の内容");
for (int i = 0; i < 6; i++) {
    printf("%d ", country[i]);
}
printf("¥n");

// 国番号 配列作る
int country_code[10][6] = {
{0,0,0,0,0,0},
{0,0,1,0,1,1},
{0,0,1,1,0,1},
{0,0,1,1,1,0},
{0,0,1,1,1,0},

```

```

{0,1,0,0,1,1},
{0,1,1,0,0,1},
{0,1,1,1,0,0},
{0,1,0,1,0,1},
{0,1,0,1,1,0},
{0,1,1,0,1,0}
};

int found = 0;
for (int i = 0; i < 10; ++i) {
    int match = 1;
    for (int j = 0; j < 6; ++j) {
        if (country_code[i][j] != country[j]) {
            match = 0;
            break;
        }
    }
    if (match) {
        printf("国コードは[%d]番です。¥n", i);
        array_[0] = i;
        found = 1;
        break;
    }
}

if (!found) {
    printf("国コードは見つかりませんでした。¥n");
}

// print jan コード
printf("jan コード");
for (int i = 0; i < 13; i++) {
    printf("%d", array_[i]);
}
printf("¥n");
}

```

## 解説及び実行結果

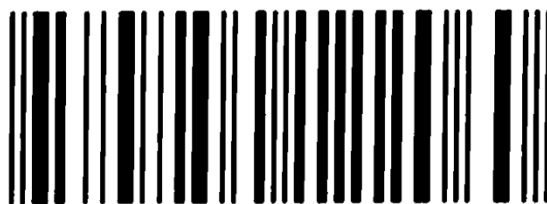
1. 撮影したバーコードの画像を閾値 125 で二値化します。

```
4 // しきい値での二値化
5 void binarize_image(int index,int threshold) {
6     int x, y, val;
7     for (y = 0; y < height[index]; y++) {
8         for (x = 0; x < width[index]; x++) {
9             val = image[index][x][y];
10
11             if (val < threshold) {
12                 image[index][x][y] = 0;
13             } else {
14                 image[index][x][y] = 255;
15             }
16         }
17     }
18 }
```

```
39 // しきい値150での二値化
40 load_image(0,"最終課題：バーコード/バーコード9784254122060.pgm");
41 copy_image(0,1);
42 binarize_image(1, 125);
43 save_image(1, "最終課題：バーコード/バーコード9784254122060_125.pgm");
```



撮影したバーコードの画像



二値化した結果

2. バーコードの、適当な 50 行目の階調値をまず配列に保存します。

```
45 // バーコードの50行目の階調値を配列Arrayに保存
46 load_image(2,"最終課題：バーコード/バーコード9784254122060_125.pgm");
47 int Array[width[2]];
48 int val,x,y=50;
49 for(x=0;x<width[2];x++){
50     val = image[2][x][y];
51     Array[x] = val;
52 }
```

```
66 // Array配列の内容を出力
67 printf("\nArray配列の内容\n");
68 for (int i = 0; i < sizeof(Array) / sizeof(Array[0]); i++) {
69     printf("%d ", Array[i]);
70 }
71 printf("\n\n");
```

Array配列の内容

```
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 0 0 0 0 0 0 0 255 255 255
255 255 255 255 255 255 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 0 0 0 0 0 0 0 255 255 255
0 0 0 0 0 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 0 0 0 0 0 0 0 255 255 255
5 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 0 0 0 0 0 0 255 255 255
5 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
5 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 0 0 0 0 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
5 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
5 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0 0 0 0 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
5 255 255 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 0 0 0 0 0 0 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
```



### 3. 黒と白の値の連続個数を数えて順に表示する

```
21 // バーコード方向への黒(0)と白(255)の値の連続個数を数えて順に表示する
22 void countConsecutiveValues(int *inputArray, int length, int *output, int outputLength) {
23     int count = 1;
24     int currentValue = inputArray[0];
25     int outputIndex = 0;
26
27     for (int i = 1; i < length; i++) {
28         if (inputArray[i] == currentValue) {
29             count++;
30         } else {
31             // printf("連続する %d の数: %d\n", currentValue, count);
32             int num;
33             num = round(count / 8.5); //最初の連続した黒と白のピクセル数で適当に割る
34             if (outputIndex < outputLength) {
35                 output[outputIndex] = num;
36                 outputIndex++;
37             } else {
38                 printf("出力配列のサイズが足りません\n");
39                 return;
40             }
41
42             currentValue = inputArray[i];
43             count = 1;
44         }
45     }
46 }
```

output配列の内容 18 8 9 7 10 23 9 15 26 8 16 8 17 24 9 8 17 7 17 16 9 24 17 7 9 8 25  
16 8 8 9 7 10 15 17 16 9 15 9 16 17 15 9 16 18 23 17 8 8 8 9 7 34 24 17 7 9 8 8 8 1

数えた結果はこうなります。2 個目から 5 個目は、最初の標準ガードに当たりますので、全部 1 ピクセルにしたいので、33 行目で平均値の 8.5 で割ります。結果は下のようになります。

output配列の内容 2 1 1 1 1 3 1 2 3 1 2 1 2 3 1 1 2 1 2 2 1 3 2 1 1 1 3  
2 1 1 1 1 1 2 2 2 1 2 1 2 2 2 1 2 2 3 2 1 1 1 1 1 4 3 2 1 1 1 1 1 1

ここまでは、バーコード 1 行で、黒(0)と白(255)の値の連続個数（ピクセル数）を数えて順に表示できました。

4. 下のバーコード表を配列にて保存します。

左・奇数					左・偶数					右			
3	2	1	1	0	1	1	2	3	0	3	2	1	1
2	2	2	1	1	1	2	2	2	1	2	2	2	1
2	1	2	2	2	2	2	1	2	2	2	1	2	2
1	4	1	1	3	1	1	4	1	3	1	4	1	1
1	1	3	2	4	2	3	1	1	4	1	1	3	2
1	2	3	1	5	1	3	2	1	5	1	2	3	1
1	1	1	4	6	4	1	1	1	6	1	1	1	4
1	3	1	2	7	2	1	3	1	7	1	3	1	2
1	2	1	3	8	3	1	2	1	8	1	2	1	3
3	1	1	2	9	2	1	1	3	9	3	1	1	2

```

86 // JANコード→バーコード 配列作る
87 int JANcode_Barcode[10][3][4] = {
88     {{3,2,1,1},{1,1,2,3},{3,2,1,1}},
89     {{2,2,2,1},{1,2,2,2},{2,2,2,1}},
90     {{2,1,2,2},{2,2,1,2},{2,1,2,2}},
91     {{1,4,1,1},{1,1,4,1},{1,4,1,1}},
92     {{1,1,3,2},{2,3,1,1},{1,1,3,2}},
93     {{1,2,3,1},{1,3,2,1},{1,2,3,1}},
94     {{1,1,1,4},{4,1,1,1},{1,1,1,4}},
95     {{1,3,1,2},{2,1,3,1},{1,3,1,2}},
96     {{1,2,1,3},{3,1,2,1},{1,2,1,3}},
97     {{3,1,1,2},{2,1,1,3},{3,1,1,2}}
98 };

```

5.output 配列の中で、ルールによって、各四つの数字をバーコード表配列に対応している行数・列数を調べます。

行数は4つの数字を表している番号で配列 array\_ に保存します。

同時に、列数は0なら、バーコード表のAパターンで、列数は1なら、バーコード表のBパターンで、最初の6つはAとBの並び方で国番号を表しますので、配列 country に保存します。

```
136     for(int n=33; n<54;n+=4){
137         int target[4] = {output[n],output[n+1],output[n+2],output[n+3]};
138         int found = 0;
139
140         for (int i = 0; i < 10; ++i) {
141             for (int j = 0; j < 3; ++j) {
142                 int match = 1;
143                 for (int k = 0; k < 4; ++k) {
144                     if (JANcode_Barcode[i][j][k] != target[k]) {
145                         match = 0;
146                         break;
147                     }
148                 }
149                 if (match) {
150                     printf("[%d,%d,%d,%d]のJANコードは[%d]行にあります\n", output[n],output[n+1],output[n+2],output[n+3],i);
151                     array_[count_array+1] = i;
152                     count_array+=1;
153
154                     found = 1;
155                     break;
156                 }
157             }
158             if (found) break;
159         }
160
161         if (!found) {
162             printf("1,3,1,2は見つかりませんでした。 \n");
163         }
164     }
```

```
100     int array_[13]; // 1~12は識別してきたjanコードを保存、0は最後の国番号を保存
101     int country[6]; // 識別してきた奇数0・偶数1を保存、奇数偶数の並びで国番号を識別する
102
103     int count_array = 0; // 配列に記入するindex
104     for(int n=4; n<27;n+=4){
105         int target[4] = {output[n],output[n+1],output[n+2],output[n+3]}; // 識別したい4桁
106         int found = 0;
107
108         for (int i = 0; i < 10; ++i) {
109             for (int j = 0; j < 3; ++j) {
110                 int match = 1;
111                 for (int k = 0; k < 4; ++k) {
112                     if (JANcode_Barcode[i][j][k] != target[k]) {
113                         match = 0;
114                         break;
115                     }
116                 }
117                 if (match) {
118                     printf("[%d,%d,%d,%d]のJANコードは[%d]行[%d]列にあります\n", output[n],output[n+1],output[n+2],output[n+3],i,j);
119                     array_[count_array+1] = i;
120                     country[count_array] = j;
121                     count_array+=1;
122
123                     found = 1;
124                     break;
125                 }
126             }
127             if (found) break;
128         }
129
130         if (!found) {
131             printf("%d,%d,%d,%dは見つかりませんでした。 \n",output[n],output[n+1],output[n+2],output[n+3]);
132         }
133     }
134 }
```

```

[1,3,1,2]のJANコードは[7]行[0]列にあります
[3,1,2,1]のJANコードは[8]行[1]列にあります
[2,3,1,1]のJANコードは[4]行[1]列にあります
[2,1,2,2]のJANコードは[2]行[0]列にあります
[1,3,2,1]のJANコードは[5]行[1]列にあります
[1,1,3,2]のJANコードは[4]行[0]列にあります
[2,2,2,1]のJANコードは[1]行にあります
[2,1,2,2]のJANコードは[2]行にあります
[2,1,2,2]のJANコードは[2]行にあります
[3,2,1,1]のJANコードは[0]行にあります
[1,1,1,4]のJANコードは[6]行にあります
[3,2,1,1]のJANコードは[0]行にあります

```

6. 5.と同じ調べ方で、国コードを調べます。

```

166 // 国番号配列の内容を出力
167 printf("国番号配列の内容");
168 for (int i = 0; i < 6; i++) {
169     printf("%d ", country[i]);
170 }
171 printf("\n");
172
173 // 国番号 配列作る
174 int country_code[10][6] = {
175     {0,0,0,0,0,0},
176     {0,0,1,0,1,1},
177     {0,0,1,1,0,1},
178     {0,0,1,1,1,0},
179     {0,1,0,0,1,1},
180     {0,1,1,0,0,1},
181     {0,1,1,1,0,0},
182     {0,1,0,1,0,1},
183     {0,1,0,1,1,0},
184     {0,1,1,0,1,0}
185 };
186
187 int found = 0;
188 for (int i = 0; i < 10; ++i) {
189     int match = 1;
190     for (int j = 0; j < 6; ++j) {
191         if (country_code[i][j] != country[j]) {
192             match = 0;
193             break;
194         }
195     }
196     if (match) {
197         printf("国コードは[%d]番です.\n", i);
198         array_[0] = i;
199         found = 1;
200         break;
201     }
202 }
203
204 if (!found) {
205     printf("国コードは見つかりませんでした.\n");
206 }

```

```

国番号配列の内容0 1 1 0 1 0
国コードは[9]番です。

```

7. 配列 `array_` の要素を出せば、jan コードの結果になります。

```
208 // print janコード
209 printf("janコード");
210 for (int i = 0; i < 13; i++) {
211     printf("%d", array_[i]);
212 }
213 printf("\n");
214 }
```

janコード 9784254122060