

```
// 「Cで学ぶデータ構造とアルゴリズム」(西原清一) オーム社, 2008
// 図4・24 (p.99) 二値画像の連結成分の抽出
// 改訂版
```

```
#include <stdio.h>
#define M 15
#define N 20
#define BLACK -1

int pic[M+1][N+2]= {
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0},
    { 0, 0, 0, 0, -1, -1, 0, 0, 0, 0, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0},
    { 0, 0, 0, -1, -1, -1, 0, 0, 0, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0},
    { 0, 0, -1, -1, -1, 0, 0, -1, -1, -1, 0, -1, -1, 0, 0, 0, -1, -1, -1, 0},
    { 0, -1, -1, -1, -1, 0, 0, -1, 0, 0, 0, -1, 0, 0, 0, -1, -1, -1, 0, 0},
    { 0, -1, -1, -1, 0, 0, 0, 0, 0, -1, 0, -1, 0, 0, -1, -1, -1, 0, 0, 0},
    { 0, 0, -1, -1, -1, 0, 0, -1, -1, 0, 0, -1, 0, 0, -1, -1, -1, 0, 0, 0},
    { 0, 0, 0, -1, -1, -1, -1, -1, 0, 0, -1, -1, -1, -1, -1, -1, -1, 0, 0, 0},
    { 0, 0, 0, 0, 0, -1, 0, 0, 0, -1, -1, -1, -1, -1, -1, 0, 0, -1, -1, 0},
    { 0, 0, 0, -1, 0, 0, 0, -1, -1, -1, 0, 0, -1, -1, 0, 0, 0, 0, -1, 0},
    { 0, 0, 0, -1, 0, 0, -1, -1, 0, 0, 0, 0, -1, -1, 0, 0, -1, -1, 0, 0},
    { 0, 0, -1, -1, -1, 0, 0, 0, 0, -1, -1, 0, -1, 0, 0, 0, -1, -1, 0, 0},
    { 0, 0, -1, -1, -1, 0, 0, 0, -1, -1, -1, 0, -1, 0, 0, -1, -1, -1, 0, 0},
    { 0, 0, 0, -1, -1, -1, -1, -1, -1, 0, -1, 0, 0, 0, -1, -1, -1, 0, 0, 0},
    { 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0}
};

int s, parent[100];

void group(int i, int j)
{
    parent[j] = i;
}

void connected_component()
{
    int i, j, k, t, sw, w[5];
    s = 0;
    for (i=1; i <= M; i++) {
        for (j=1; j <= N; j++) {
            if (pic[i][j] == BLACK) {
```

```

        sw = 0; t = 0;
        w[1] = pic[i-1][j+1];    w[2] = pic[i-1][j];
        w[3] = pic[i-1][j-1];    w[4] = pic[i][j-1];
        for (k=1; k <= 4; k++)
            if (w[k] != 0) {sw = 1; if (t == 0 || t > w[k]) t = w[k];}
        if (sw == 0) pic[i][j] = ++s;
        else {pic[i][j] = t;
        for (k=1; k <= 4; k++)
            if (w[k] != 0 && t != w[k]) group(t,w[k]);}
    }
}
}

main()
{    int i,j;
    connected_component();
    for (i=0; i <= M; i++) {
        for (j=0; j <= N+1; j++) {
            if (pic[i][j] == 0) printf(" .");
            else printf("%2d",pic[i][j]);
        }
        printf("\n");
    }
    printf("*parent table\n");
    for (i=1; i<=s; i++) printf("parent[%2d] = %2d\n",i,parent[i]);
    j=0;
    for (i=1; i<=s; i++) if (parent[i] == 0) j++;
    printf("The number of components of the binary image is %d.\n\n",j);
}

```