Name : Muhammad Daffa Khairi
ID : 2246176007

Exercise A05

- **Goal**

  Complete the function "**calc_chain_code()**" in chain.c. Apply **labeling** and **contour detection** to binary image "**sample.pbm**".

- **Output**

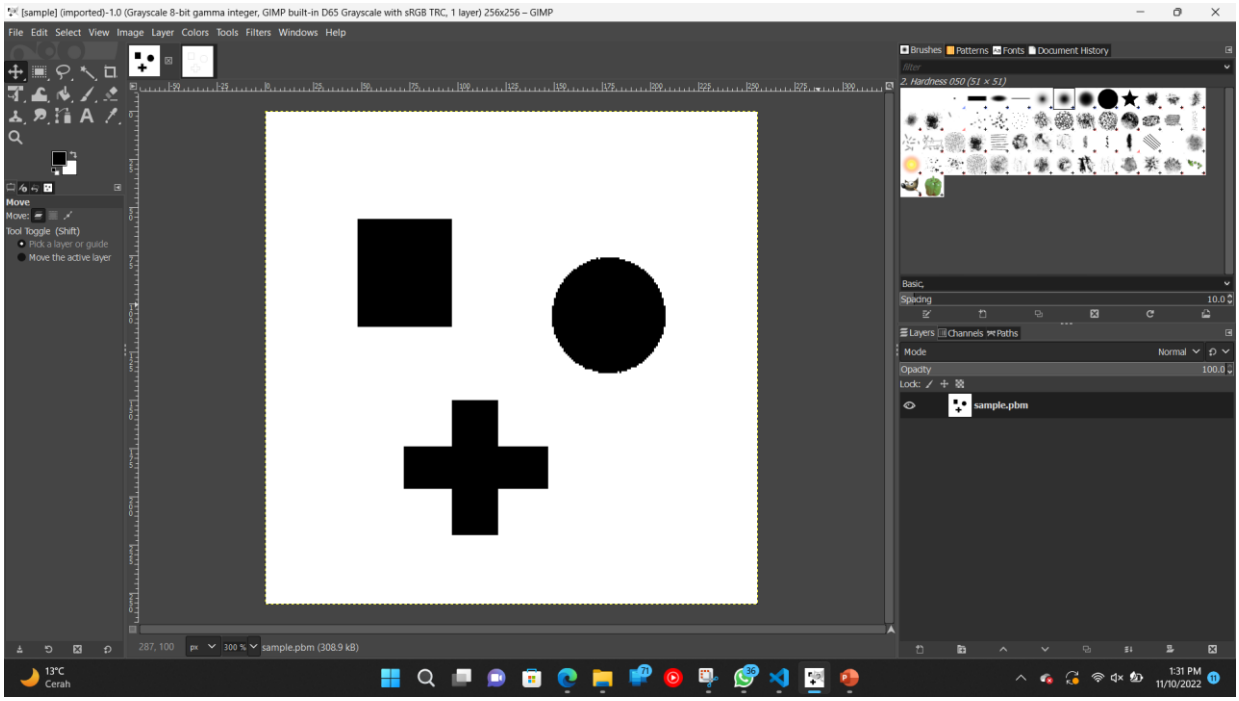

Figure 1. Original Image

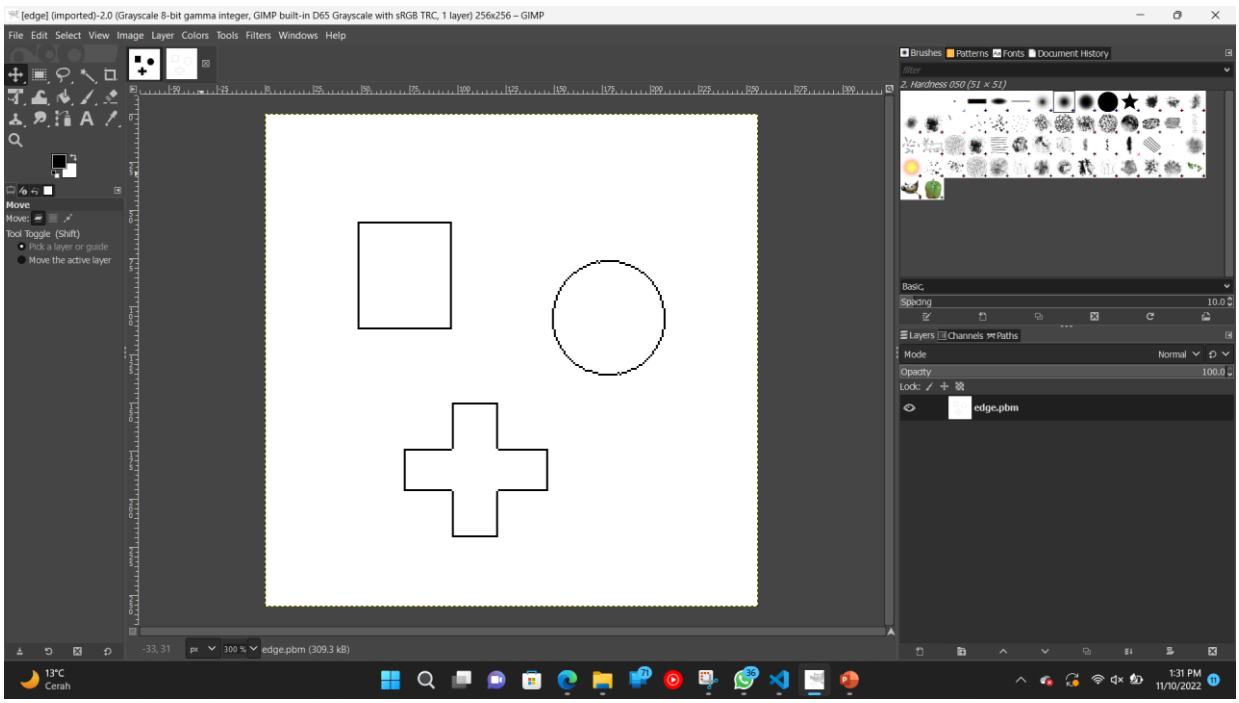

Figure 2. Labeling and Contour Detection Result

- **Code Specification**

Filename : **chain.c**	
Specification of Function	
Name	calc_chain_code
Arguments	Bitmap* labelIM (Input Binary Image) int label_no (number of label) Bitmap* edgeIM (Output Binary Image)
Return Value	int
Summary	Chain code

- **Code Algorithm**

Algorithm detect_contour

Variables :
- *x* structure of input label image
- *ln* number of label
- *y* structure of output contour image

Initialize: Set 0 to all pixel of *y*
For each label *l* in the image
  If detect pixel of label *l* by raster-scan **then**
    Set current point (*cy, cx*) to start point (*sy, sx*)
  Set 5 to chain code *cn*
  Do
    Do
      Set target point (*ty, tx*) to direction *cn* of current point (*cy, cx*)
      If *x*(*ty,tx*) is not equal to l then
        Set next direction (Fig.A) to *cn*
      While *x*(*ty,tx*) is not equal to *l*
      Record chain code *cn*
      Set target point (ty, tx) to current point (cy, cx)
      y(*cy, cx*) ← 1, ln ← ln + 1
      Set next start code (Table A) to *cn*
  while current point (cy, cx) is not equal to start point (sy, sx)

- **Code**

1. chain.c

```
int calc_chain_code(Bitmap* labelIM, int label_no, Bitmap* edgeIM)
{
        /*******************************************/
    int chain_code[8][2] = {
        { 0, 1},
        {-1, 1},
        {-1, 0},
        {-1,-1},
        { 0,-1},
        { 1,-1},
        { 1, 0},
        { 1, 1} };
    int next_code_no[8] = { 7,7, 1, 1, 3, 3, 5, 5}; /* next start code */
        /*******************************************/

    int x, y, xx, yy, start_x, start_y;
    int code_no, found, code_length;
    char code_str[CODE_LENGTH_MAX];

    for (y = 0; y < labelIM->height; y++)
        for (x = 0; x < labelIM->width; x++)
            if (labelIM->map[y * labelIM->width + x] == label_no) goto
LABEL_FOUND;

LABEL_FOUND:
    start_x = x;
    start_y = y;
    code_no = 5;
    code_length = 0;
    do {
        found = 0;
        do {
            xx = x + chain_code[code_no][1];
            yy = y + chain_code[code_no][0];
            if ((xx >= 0 && xx < labelIM->width && yy >= 0 && yy < labelIM-
>height) && labelIM->map[yy * labelIM->width + xx] == label_no)
                found = 1;
            else {
                code_no++;
                if (code_no > 7) code_no = 0;
            }
        } while (found == 0);
        sprintf(&code_str[code_length], "%d", code_no);
        code_length++;
        x = xx;
        y = yy;
        edgeIM->map[y * edgeIM->width + x] = 1;
        code_no = next_code_no[code_no];
    } while (!(x == start_x && y == start_y));
    printf("(%d,%d) %s\n", start_x, start_y, code_str);

    return code_length;
}
```