

[Assignment 4]

Fill Your Heart

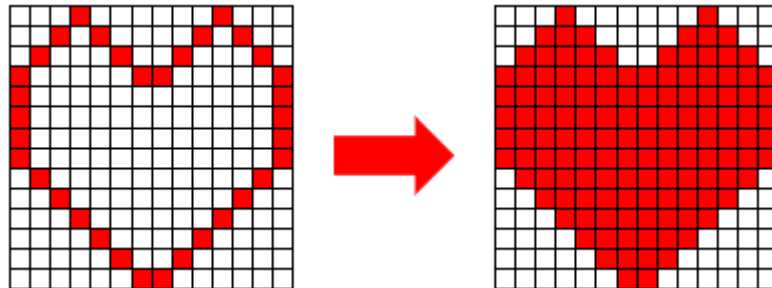


Figure 1

[Description]

Given a lined heart, you should write a program that fills inside of it with colors in recursive manner. In this program, a pixel whose value is 0 means that it is non-colored, and 1 means it is colored (see Figure 2). The details of instruction are presented as below.

- 1) Get a starting pixel (row and column), and a lined heart with 14x14 size from input.txt file using file I/O. (see input.txt picture in Examples)
 - A. The two integers at the first line of input.txt are the starting row and column. The starting point must be inside of the shape.
 - B. The rest of the file is a shape of the figure, a whole 2D array
- 2) Print the original input as below

```
=====Empty Heart=====
[0][0][0][1][0][0][0][0][0][1][0][0][0][0]
[0][0][1][0][1][0][0][0][0][1][0][1][0][0]
[0][1][0][0][0][1][0][0][1][0][0][0][1][0]
[1][0][0][0][0][0][1][1][0][0][0][0][0][1]
[1][0][0][0][0][0][0][0][0][0][0][0][0][1]
[1][0][0][0][0][0][0][0][0][0][0][0][0][1]
[1][0][0][0][0][0][0][0][0][0][0][0][0][1]
[1][0][0][0][0][0][0][0][0][0][0][0][0][1]
[0][1][0][0][0][0][0][0][0][0][0][0][1][0]
[0][0][1][0][0][0][0][0][0][0][0][1][0][0]
[0][0][0][1][0][0][0][0][0][0][1][0][0][0]
[0][0][0][0][1][0][0][0][0][1][0][0][0][0]
[0][0][0][0][0][1][0][0][0][0][0][0][0][0]
[0][0][0][0][0][0][1][0][0][1][0][0][0][0]
[0][0][0][0][0][0][0][1][0][1][0][0][0][0]
[0][0][0][0][0][0][0][0][1][1][0][0][0][0]
```

Figure 2

- 3) Design and implement a recursive function which fills the inside of the heart with 1. The complete result is this.
 - A. From the starting pixel, check whether the current pixel is non-colored (0).

- B. If it is non-colored, fill the pixel with color (1).
- C. Visit its neighbors.
 - i. You can move to all 4 directions (up, down, right, and left)
- D. Repeat A to C until the figure is fully colored.

```

=====Filled Heart=====
[0][0][0][1][0][0][0][0][0][1][0][0][0]
[0][0][1][1][1][0][0][0][0][1][1][1][0][0]
[0][1][1][1][1][1][0][0][1][1][1][1][1][0]
[1][1][1][1][1][1][1][1][1][1][1][1][1][1]
[1][1][1][1][1][1][1][1][1][1][1][1][1][1]
[1][1][1][1][1][1][1][1][1][1][1][1][1][1]
[1][1][1][1][1][1][1][1][1][1][1][1][1][1]
[1][1][1][1][1][1][1][1][1][1][1][1][1][1]
[0][1][1][1][1][1][1][1][1][1][1][1][1][0]
[0][0][1][1][1][1][1][1][1][1][1][1][0][0]
[0][0][0][1][1][1][1][1][1][1][1][1][0][0]
[0][0][0][0][1][1][1][1][1][1][1][0][0][0]
[0][0][0][0][0][1][1][1][1][1][1][0][0][0]
[0][0][0][0][0][0][1][1][1][1][1][0][0][0]
[0][0][0][0][0][0][0][1][1][1][0][0][0][0]
[0][0][0][0][0][0][0][0][1][1][0][0][0][0]

```

- 4) Save the result into "output.txt" file using file I/O.
- 5) Your program also must work with any other closed figures such as squares, circles, etc.
You can test your code with given test sets.
- 6) You can check the progress of your program by using "printArray" function below.

```

void print_array(int A[14][14]) {
    for(int i=0; i<14; i++) {
        for(int j=0; j<14; j++) {
            printf("%d", A[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}

```

[Examples]

```

=====Empty Heart=====
[0][0][0][1][0][0][0][0][0][1][0][0][0]
[0][0][1][1][0][0][0][0][1][0][1][0][0]
[0][1][0][0][0][1][0][0][1][0][0][0][1][0]
[1][0][0][0][0][0][1][1][0][0][0][0][0][1]
[1][0][0][0][0][0][0][0][0][0][0][0][0][1]
[1][0][0][0][0][0][0][0][0][0][0][0][0][1]
[1][0][0][0][0][0][0][0][0][0][0][0][0][1]
[1][0][0][0][0][0][0][0][0][0][0][0][0][1]
[0][1][0][0][0][0][0][0][0][0][0][1][0]
[0][0][1][0][0][0][0][0][0][0][1][0][0]
[0][0][0][1][0][0][0][0][0][0][1][0][0]
[0][0][0][0][1][0][0][0][0][1][0][0][0]
[0][0][0][0][0][1][0][0][1][0][0][0][0]
[0][0][0][0][0][0][1][1][0][0][0][0][0]
[0][0][0][0][0][0][0][1][1][0][0][0][0]
Starting from [8, 6]

```

At the beginning

```

=====Filled Heart=====
[0] [0] [0] [1] [0] [0] [0] [0] [0] [0] [1] [0] [0] [0]
[0] [0] [1] [1] [1] [0] [0] [0] [0] [1] [1] [1] [0] [0]
[0] [1] [1] [1] [1] [1] [0] [0] [1] [1] [1] [1] [1] [0]
[1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]
[1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]
[1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]
[1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]
[1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1]
[0] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [0] [0]
[0] [0] [1] [1] [1] [1] [1] [1] [1] [1] [1] [1] [0] [0]
[0] [0] [0] [1] [1] [1] [1] [1] [1] [1] [1] [0] [0] [0]
[0] [0] [0] [0] [1] [1] [1] [1] [1] [1] [0] [0] [0] [0]
[0] [0] [0] [0] [0] [1] [1] [1] [1] [0] [0] [0] [0] [0]
[0] [0] [0] [0] [0] [0] [1] [1] [0] [0] [0] [0] [0] [0]

```

The result

```

input.txt - 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)
8 6
00010000001000
001010000010100
010001001000010
100000110000001
100000000000001
100000000000001
100000000000001
100000000000001
010000000000010
001000000000100
000100000001000
000010000010000
0000010000010000

```

input.txt

```

output.txt - 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)
00010000001000
001110000011100
01111100111110
111111111111111
111111111111111
111111111111111
111111111111111
111111111111111
111111111111111
011111111111110
001111111111100
000111111111000
000011111110000
000001111100000

```

output.txt

[Rating]

- Total points =100.
- 1) Print out the starting row, column, and the original input (20 points).
- 2) Print out the filled result (30 points).
- 3) Save the result into output file (20 points).
- 4) Use a recursive function (30 points).

[CAUTION]

- Delay penalty: After due date, 15 points will be deducted for every single day. And submission will not be entertained after 3rd day.
- Submit the successfully compiled source code on I-Campus.

- If you have a question about assignment, send an e-mail to skwi2014@naver.com. Make sure write your name and the point what you want to ask.
- Just printing the answer only using printf function will be meaningless (no points), so don't try it.