### Level19.5

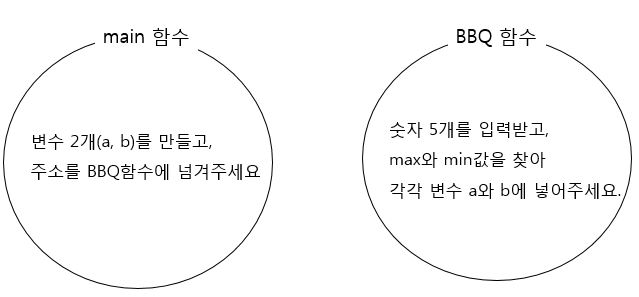
고생많으셨습니다! DAT / direct / pattern까지 for의 기법들을 모두 배우셨습니다.

이제부터 반복하면서 다양한 for문 문제를 풀게 됩니다.

**for문 훈련문제를 풀 때 꼭 연습장에 설계를 한 후 풀어주세요**

## Level19.5 함수와 포인터 [난이도 : 1]

**문제 1번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1061)]



**※ main 함수에서 a, b를 출력 해주세요.**

## 입력 예제

4 2 5 3 8

## 출력 결과

a=8

b=2

**[소스 코드]**

#include <iostream>

using std::endl;

using std::cout;

using std::cin;

#define MAX 5

void BBQ(int\* \_max,int\* \_min)

{

int arr[MAX] = {};

for (int i = 0; i < MAX; i++)

{

cin >> arr[i];

if (\*\_max < arr[i])

{

\*\_max = arr[i];

}

if (\*\_min > arr[i])

{

\*\_min = arr[i];

}

}

}

int main()

{

int a = -10000000000;

int b = 10000000000;

BBQ(&a, &b);

cout << "a=" << a << endl;

cout << "b=" << b << endl;

}

## Level19.5 안정적인 세포 판별 [난이도 : 4]

**문제 2번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1061)]

5x4 배열에 0 또는 1로 구성된 숫자들을 입력받아주세요.

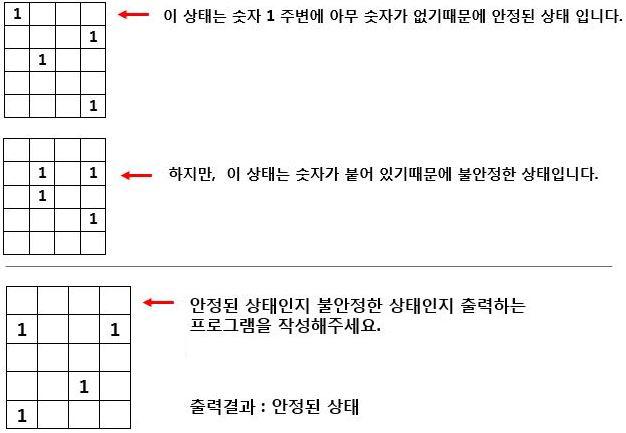
입력받은 배열에 있는 숫자 1은 세포를 뜻합니다.

세포 1주변에 아무것도 없으면 세포들이 안정된 상태입니다.

즉, 8방향 모두 아무런 숫자가 없어야 합니다.

아래 예제를 참고하여,

입력받은 세포상태가 "**안정된 상태**"인지 **"불안정한 상태"**인지 출력 하세요.



## 입력 예제

0 0 0 0

1 0 0 0

0 0 1 0

0 0 0 0

1 0 0 1

## 출력 결과

안정된 상태

**[소스 코드]**

#include <iostream>

using std::endl;

using std::cout;

using std::cin;

#define MAX 5

#define COL 5

#define ROW 4

struct TrueIndex

{

int col;

int row;

};

bool check(const TrueIndex& \_index, bool(\*\_arr)[ROW])

{

int index\_i = \_index.col;

int index\_j = \_index.row;

if (index\_i - 1 >= 0 && index\_i + 1 < COL && index\_j - 1 >= 0 && index\_j + 1 < ROW)

{

if (index\_i - 1 >= 0 && \_arr[index\_i - 1][index\_j] == 1)

{

return true;

}

if (index\_i - 1 >= 0 && index\_j - 1 >= 0 && \_arr[index\_i - 1][index\_j - 1] == 1)

{

return true;

}

if (index\_i - 1 >= 0 && index\_j + 1 < ROW && \_arr[index\_i - 1][index\_j + 1] == 1)

{

return true;

}

if (index\_j - 1 >= 0 && \_arr[index\_i][index\_j - 1] == 1)

{

return true;

}

if (index\_j + 1 < ROW && \_arr[index\_i][index\_j + 1] == 1)

{

return true;

}

if (index\_i + 1 < COL && index\_j - 1 >= 0 && \_arr[index\_i + 1][index\_j - 1] == 1)

{

return true;

}

if (index\_i + 1 < COL && index\_j + 1 < ROW && \_arr[index\_i + 1][index\_j + 1] == 1)

{

return true;

}

}

return false;

}

int main()

{

bool arr[COL][ROW] = {};

TrueIndex true\_arr[COL \* ROW] = {};

int true\_index = 0;

for (int i = 0; i < COL; i++)

{

for (int j = 0; j < ROW; j++)

{

cin >> arr[i][j];

if (arr[i][j] == 1)

{

true\_arr[true\_index].col = i;

true\_arr[true\_index++].row = j;

}

}

}

bool flag = false;

for (int i = 0; i < true\_index; i++)

{

flag = check(true\_arr[i], arr);

if (flag == true)

break;

}

if (flag == true)

{

cout << "불안정한 상태";

}

else

{

cout << "안정된 상태";

}

return 0;

}

## Level19.5 핸드폰 비밀번호 순서 [난이도 : 5]

**문제 3번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1061)]

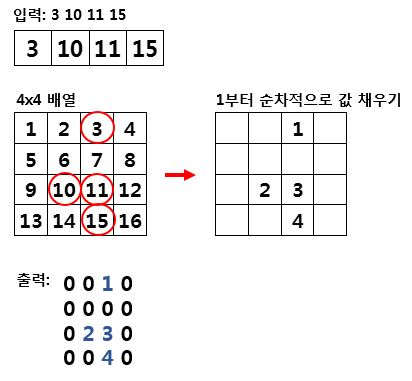
**1~16 사이의 숫자 4개를 입력** 받고 배열에 채워주세요.

총 16칸짜리인 **4x4 배열**을 만들어 주세요.

4x4 배열에 아래와 같이 번호를 매긴다고 했을때

입력받은 숫자 4개에 해당하는 번호에 **값을 1부터 순차적으로 채워준 후 출력** 해주세요.

**아래 예제를 참고하여 소스코드를 작성해주세요.**

****

## 입력 예제

3 10 11 15

## 출력 결과

0 0 1 0

0 0 0 0

0 2 3 0

0 0 4 0

**[소스 코드]**

#include <iostream>

using std::endl;

using std::cout;

using std::cin;

#define MAX 4

#define COL 5

#define ROW 4

int main()

{

int arr[MAX] = {};

for (int i = 0; i < MAX; i++)

{

cin >> arr[i];

if (arr[i] <= 0 || arr[i] >= 17)

{

--i;

continue;

}

}

int arr2[MAX][MAX] = {};

int cnt = 1;

for (int i = 0; i < MAX; i++)

{

int index\_i = arr[i] / MAX;

int index\_j = (arr[i] % MAX) == 0 ? 3 : (arr[i] % MAX) - 1;

arr2[index\_i][index\_j] = cnt++;

}

return 0;

}

## Level19.5 가로세로 색칠하기 [난이도 : 2]

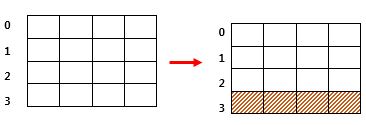
**문제 4번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1061)]

가로 또는 세로를 색칠하는 프로그램을 짜야 합니다.

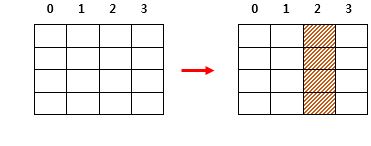
4x4 배열 하나를 준비 해 주세요.

만약

**G 3**이라고 입력 받으면 **가로 3번줄**이 색칠 됩니다.



**S 2**를 입력 받으면 **세로 2번줄**이 색칠 됩니다.



이러한 규칙으로 **3개의 명령어를 입력 받고 결과를 출력** 하세요.

**ex)**

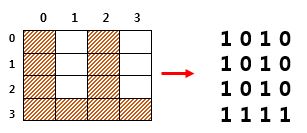
입력:

**G 3**

**S 2**

**S 0**

출력:



## 입력 예제

G 3

S 2

S 0

## 출력 결과

1 0 1 0

1 0 1 0

1 0 1 0

1 1 1 1

**[소스 코드]**

#include <iostream>

using std::endl;

using std::cout;

using std::cin;

#define MAX 4

#define COL 5

#define ROW 4

int main()

{

char input\_c = ' ';

int input\_index = 0;

bool result[MAX][MAX] = {};

for (int loop = 0; loop < 3; loop++)

{

cin >> input\_c >> input\_index;

switch (input\_c)

{

case 'G':

for (int i = 0; i < MAX; i++)

{

result[input\_index][i] = true;

}

break;

case 'S':

for (int i = 0; i < MAX; i++)

{

result[i][input\_index] = true;

}

break;

}

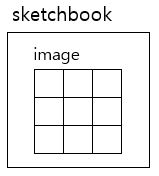
}

return 0;

}

## Level19.5 모양넣기 [난이도 : 5]

**문제 5번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1061)]

왼쪽 그림과 같은 **sketchbook 구조체를 정의**하고, **구조체 변수 1개**를 만들어 주세요.

이미지에 들어갈 내용을 **구조체 변수안 image char배열에 입력받고,**

어떤 무늬로 구성되어 있는지 **알파벳 순서대로 출력**하세요.

(입력되는 문자는 모두 대문자입니다)

ex)

**[입력] [출력]**

ATK AGKT

AAA

TTG

## 입력 예제

BBB

BCD

DZZ

## 출력 결과

BCDZ

**[소스 코드]**

#include <iostream>

using std::endl;

using std::cout;

using std::cin;

#define MAX 3

#define COL 5

#define ROW 4

#define CMAX 200

struct Index

{

int i;

int j;

};

struct sketchbook

{

char image[MAX][MAX];

};

int main()

{

sketchbook book;

bool result[CMAX] = {};

char result\_ch[MAX \* MAX] = {};

int index\_cnt = 0;

for (int i = 0; i < MAX; i++)

{

for (int j = 0; j < MAX; j++)

{

cin >> book.image[i][j];

}

}

for (int i = 0; i < MAX; i++)

{

for (int j = 0; j < MAX; j++)

{

if (result[book.image[i][j]] == false)

{

result[book.image[i][j]] = true;

result\_ch[index\_cnt++] = book.image[i][j];

}

}

}

for (int i = 0; i < index\_cnt; i++)

{

cout << result\_ch[i];

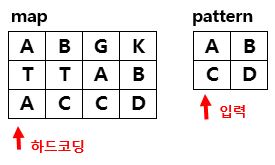
}

return 0;

}

## Level19.5 비밀 위치 찾기 [난이도 : 4]

**문제 6번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1061)]



3x4 **map배열은** 위와 같이 **하드코딩** 하고,

2x2 **pattern배열을 입력** 받아주세요.

그리고 map배열에 **pattern**라는 패턴 배열이 **존재하는지 확인**하고

**몇개**인지 출력 하면 됩니다.

만약, pattern이 존재하고 1개가 발견되었다면 **"발견(1개)"** 출력

만약, pattern이 없다면 **"미발견"** 출력

## 입력 예제

AB

CD

## 출력 결과

발견(1개)

**[소스 코드]**

#include <iostream>

using std::endl;

using std::cout;

using std::cin;

#define MAX 2

#define COL 3

#define ROW 5

#define CMAX 200

int main()

{

char map[COL][ROW] = {"ABGK","TTAB","ACCD"};

char pattern[MAX][MAX] = {};

for (int i = 0; i < MAX; i++)

{

for (int j = 0; j < MAX; j++)

{

cin >> pattern[i][j];

}

}

int cnt = 0;

for (int i = 0; i < COL; i++)

{

for (int j = 0; j < ROW; j++)

{

if (map[i][j] ==pattern[0][0])

{

if (map[i][j + 1] == pattern[0][1] && map[i + 1][j] == pattern[1][0]

&& map[i + 1][j + 1] == pattern[1][1])

{

cnt++;

}

}

}

}

if (cnt > 0)

{

cout << "발견(" << cnt << "개)";

}

else

{

cout << "미발견";

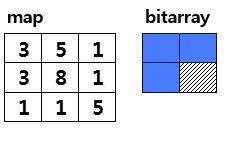
}

return 0;

}

## Level19.5 마스킹하고 난뒤 [난이도 : 5]

**문제 7번** [[숙제 목록보기](http://quest.mincoding.co.kr/contest.php?cid=1061)]



2x2 size의 **bitarray 배열을 입력** 받고, **map 배열을 하드코딩** 하세요.

bitarray를 map의 (0,0)에 masking하면 나오는 값은

 이고 합은 11입니다.

bitarray를 map에다 masking 후

**합**을 구했을 때 **가장 큰 값이 나오는 좌표**를 출력 하세요.

## 입력 예제

1 1

1 0

## 출력 결과

(0,1)

**[소스 코드]**

#include <iostream>

using std::endl;

using std::cout;

using std::cin;

#define MAX 2

#define COL 3

#define ROW 3

#define CMAX 200

struct Index

{

int x;

int y;

};

int check(int(\*\_map)[ROW], bool(\*\_bit)[MAX], const Index& \_index)

{

int sum = 0;

if (\_bit[0][0] == true)

{

sum += \_map[\_index.y][\_index.x];

}

if (\_bit[0][1] == true&&\_index.x+1 < ROW)

{

sum += \_map[\_index.y][\_index.x + 1];

}

if (\_bit[1][0] == true&&\_index.y+1<COL)

{

sum += \_map[\_index.y + 1][\_index.x];

}

if (\_bit[1][1] == true && \_index.y + 1 < COL && \_index.x + 1 < ROW)

{

sum += \_map[\_index.y + 1][\_index.x + 1];

}

return sum;

}

int main()

{

int map[COL][ROW] = { 3,5,1,

3,8,1,

1,1,5 };

bool bitarray[MAX][MAX] = {};

for (int i = 0; i < MAX; i++)

{

for (int j = 0; j < MAX; j++)

{

cin >> bitarray[i][j];

}

}

int sum = 0;

int max = -100000000;

Index max\_index;

for (int i = 0; i < COL; i++)

{

for (int j = 0; j < ROW; j++)

{

Index index{ j,i };

sum = check(map, bitarray, index);

if (max < sum)

{

max = sum;

max\_index = index;

}

sum = 0;

}

}

cout << "(" << max\_index.y << "," << max\_index.x << ")";

return 0;

}