

## (훈련반2) Level27

링크드리스트 네번째 입니다.

특정 **노드를 삭제하는 것과, 노드를 추가**하는 것을 연습합니다.

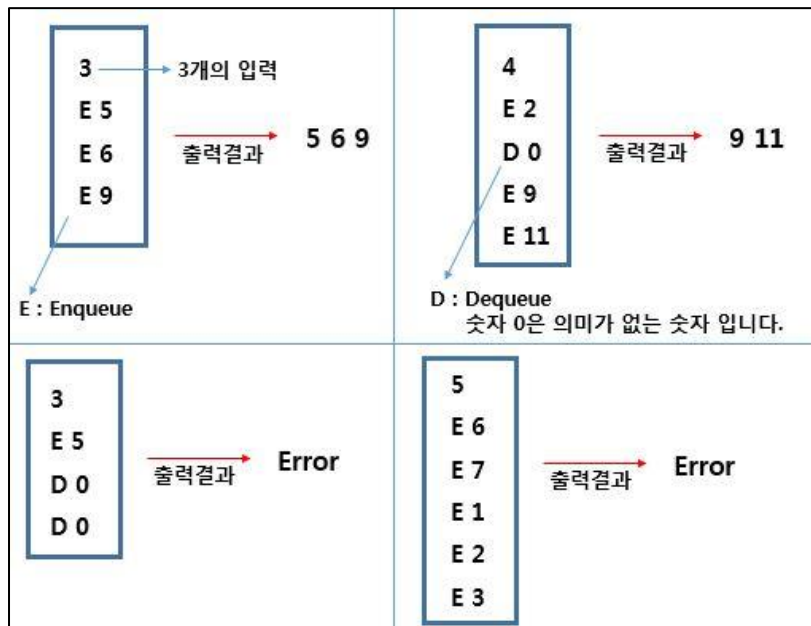
링크드리스트는 next->next->next ... 이렇게 next가 여러개 있을 때 가리키고 있는 노드가 이 노드인지, 다음노드인지 혼란이 자주 옵니다. 헷갈리지 않도록 익숙해져야합니다.

## Level27 큐가 비었을때

문제 1번 [\[숙제 목록보기\]](#)

4칸짜리 큐를 링크드리스트로 만들어주세요.

그리고 아래와 같이 입력 받았을때 최종 큐의 결과를 출력 하세요.



## 입력 예제

3  
E 5  
E 6  
E 9

## 출력 결과

5 6 9

```

#include <iostream>
#define _CRTDBG_MAP_ALLOC
#include <stdlib.h>
#include <crtDBG.h>
using namespace std;

template <typename T>
class MyQueue
{
public:
    MyQueue()
        : head(nullptr)
        , tail(nullptr)
    {
    }
    struct Node
    {
    public:
        Node()
            : data(0)
            , pNext(nullptr)
        {
        }
        Node(T _data)
            : data(_data)
            , pNext(nullptr)
        {
        }

        T data;
        Node* pNext;
    };

    void Enqueue(int _data)
    {
        if (head == nullptr)
        {
            head = new Node(_data);
            tail = head;
        }
        else
        {
            tail->pNext = new Node(_data);
            tail = tail->pNext;
        }
    }
}

```

```

T Dequeue()
{
    T ret = head->data;
    Node* temp = head;
    head = head->pNext;
    delete temp;
    return ret;
}

void Print()
{
    Node* temp = head;
    while (temp != nullptr)
    {
        cout << temp->data << " ";
        temp = temp->pNext;
    }
}

void Delet()
{
    Node* temp1 = head;
    Node* temp2 = nullptr;
    while (temp1 != nullptr)
    {
        temp2 = temp1->pNext;
        if (temp1 == head)
            head = nullptr;
        delete temp1;
        temp1 = temp2;
    }
}

private:
    Node* head;
    Node* tail;
};

```

```
// 4칸짜리 큐를 링크드 리스트로 구현
```

```
int main(void)
```

```
{
```

```
    MyQueue<int>* que = new MyQueue<int>();
```

```
    int input = 0;
```

```
    cin >> input;
```

```
    char* cmds = new char[input];
```

```
    int* vals = new int[input];
```

```
    for (int i = 0; i < input; ++i)
```

```
        cin >> cmds[i] >> vals[i];
```

```
    for (int k = 0; k < input; ++k)
```

```
    {
```

```
        if (cmds[k] == 'E')
```

```
            que->Enqueue(vals[k]);
```

```
        else if (cmds[k] == 'D')
```

```
            que->Dequeue();
```

```
        else
```

```
            __noop;
```

```
    }
```

```
    que->Print();
```

```
    que->Delet();
```

```
    delete que;
```

```
    que = nullptr;
```

```
    delete[] cmds;
```

```
    cmds = nullptr;
```

```
    delete[] vals;
```

```
    vals = nullptr;
```

```
    _CrtDumpMemoryLeaks();
```

```
    return 0;
```

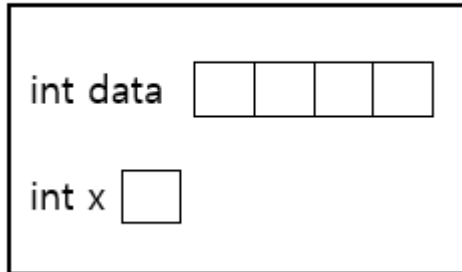
```
}
```

# Level27 구조체 포인터

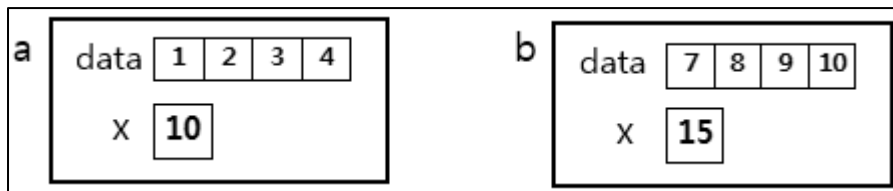
문제 2번 [숙제 목록보기]

ABC 타입 구조체는 다음과 같습니다.

ABC



ABC 타입 구조체 변수 a, b를 만들고, 다음과 같이 값을 초기화 해주세요.



구조체 포인터 p, g를 만들고,

p는 a를 가르키도록 하고

g는 b를 가르키도록 해주세요.

이제 p와 g를 이용해서 p와 g가 가르키고 있는 곳의 값들을 출력 해주세요.

## 출력 결과

```
1 2 3 4
10
7 8 9 10
15
```

```

#include <iostream>
using namespace std;

const int dataSize = 4;

struct ABC
{
public:
    ABC()
        : data{}
        , x(0)
    {

    }

    void insertArr(int* arr)
    {
        for (int i = 0; i < dataSize; ++i)
            data[i] = arr[i];
    }

    int data[dataSize];
    int x;
};

int main(void)
{
    ABC a;
    ABC b;
    int arrA[dataSize] = { 1,2,3,4 };
    int arrB[dataSize] = { 7,8,9,10 };
    a.insertArr(arrA);
    b.insertArr(arrB);
    a.x = 10;
    b.x = 15;

    ABC* p = nullptr;
    ABC* g = nullptr;
    p = &a;
    g = &b;

    for (int i = 0; i < dataSize; ++i)
        cout << p->data[i] << " ";
    cout << endl << p->x << endl;

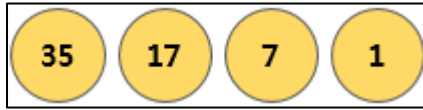
    for(int k = 0; k < dataSize; ++k)
        cout << g->data[k] << " ";

```

```
cout << endl << g->x;  
  
return 0;  
}
```

## Level27 동전교환기 (난이도 : ★☆☆)

문제 3번 [숙제 목록보기]



짜리 동전이 있습니다.

동전교환할 금액을 입력 받아주세요.

가장 큰 금액의 동전부터 교환해주려고 합니다.

각 동전마다 몇개의 동전으로 교환 할 수 있는지 출력 하세요.

ex)

입력: 170

35:4개

17:1개

7:1개

1:6개

### 입력 예제

170

### 출력 결과

35:4개

17:1개

7:1개

1:6개



```
#include <iostream>
using namespace std;

const int coinCnt = 4;

int main(void)
{
    int coins[coinCnt] = { 35,17,7,1 };
    int cnts[coinCnt] = {};
    int input = 0;
    cin >> input;

    for (int i = 0; i < coinCnt; ++i)
    {
        if (input > coins[i])
        {
            cnts[i] = input / coins[i];
            input -= (coins[i] * cnts[i]);
        }
    }

    for (int k = 0; k < coinCnt; ++k)
        cout << coins[k] << ":" << cnts[k] << "개\n";

    return 0;
}
```

# Level27 여러종류의 알파벳

문제 4번 [[숙제](#) [목록보기](#)]

한문장을 입력 받으세요.(최대 10글자)

몇가지 종류의 알파벳이 존재하는지 출력 하세요.

알파벳은 A ~ Z 까지 나올 수 있습니다

ex)

입력: AGGABC

출력: 4종류

(A, G, B, C 이렇게 네 종류의 알파벳이 있으므로 4 입니다)

## 입력 예제

AGGABC

## 출력 결과

4종류

```
#include <iostream>
using namespace std;

const int inputSize = 10;
const int alphaSize = 25; //65(A) ~ 90(Z)
const int offset = 65;
char input[inputSize + 1] = {};
char isExsit[alphaSize] = {};

int main(void)
{
    cin >> input;
    for (int i = 0; i < strlen(input); ++i)
    {
        int idx = input[i] - offset;
        ++isExsit[idx];
    }

    int cnt = 0;
    for (int k = 0; k < alphaSize; ++k)
    {
        if (isExsit[k] != 0)
            ++cnt;
    }

    cout << cnt << "종류";

    return 0;
}
```

# Level27 병정개미 문제

문제 5번 [숙제 목록보기]

4x4 배열에 병정개미 4마리를 입력하여 배치 합니다.

숫자 1이 병정개미라고 했을 때

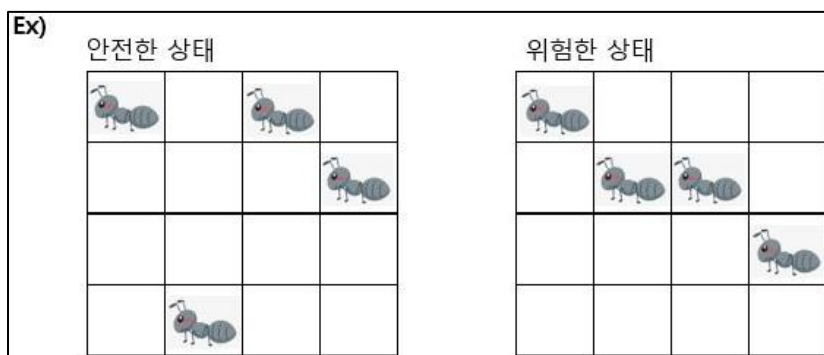
만약, 아래와 같이 입력하면 붙어있는 개미가 있어 위험한 상태 입니다.

1 0 0 0

0 1 1 0

0 0 0 1

0 0 0 0



4x4 배열에 병정개미 4마리를 배치한 값을 입력받고,

개미의 상태가 안전하면 "안전한 상태",

안전하지 않으면 "위험한 상태"라고 판단하여 출력 해주는 프로그램을 만들어 주세요.

## 입력 예제

1 0 0 0

0 1 1 0

0 0 0 1

0 0 0 0

## 출력 결과

위험한 상태

```

#include <iostream>
using namespace std;

const int mapSize = 4;
int map[mapSize][mapSize] = {};

struct Ant
{
public:
    Ant()
        : x(0)
        , y(0)
    {
    }
    void place(int _x, int _y)
    {
        x = _x;
        y = _y;
    }
    bool isSafe(Ant other)
    {
        int gap = 0;
        if (other.x == x)
        {
            gap = other.y - y;
            if (gap == (-1) || gap == 1)
                return false;
        }
        else if (other.y == y)
        {
            gap = other.x - x;
            if (gap == (-1) || gap == 1)
                return false;
        }
        else
            return true;
    }

    int x;
    int y;
};

// 붙어있는 상태는 위험한 상태
// 만약 x 축이 같다면 y 의 값이 1 차이나면 위험
// y 축이 같다면 x 의 값이 1 차이나면 위험
int main(void)
{

```

```

int antCnt = 0;
int input = 0;
for (int i = 0; i < mapSize * mapSize; ++i)
{
    cin >> input;
    map[i / mapSize][i % mapSize] = input;
    if (input != 0)
        ++antCnt;
}

Ant* ants = new Ant[antCnt]();
int findCnt = 0;
int idx = 0;
while (findCnt < antCnt) // 모든 개미의 위치를 찾으면 반복문을 멈춘다.
{
    int y = idx / mapSize;
    int x = idx % mapSize;
    if (map[y][x] != 0)
    {
        ants[findCnt].place(y, x);
        ++findCnt;
    }
    ++idx;
}

bool isSafe = true;
for (int m = 0; m < antCnt - 1; ++m)
{
    for (int n = m + 1; n < antCnt; ++n)
    {
        isSafe = ants[m].isSafe(ants[n]);
        if (!isSafe)
            break;
    }
    if (!isSafe)
        break;
}

if (isSafe)
    cout << "안전한 상태";
else
    cout << "위험함 상태";

delete[] ants;

return 0;
}

```

# Level27 소문자로 바꾸기

문제 6번 [[숙제](#) [목록보기](#)]

4x10 char 배열에 4문장을 대문자로 입력 받으세요.(최대 10글자)

가장 긴 문장과 가장 짧은 문장을 찾아내고 모두 소문자로 바꾸어주세요.

(가장 긴문장과 짧은 문장은 한개씩 존재합니다)

ex)

[입력]

ABC

BBBQ

BT

JOW

[출력]

ABC

bbbq

bt

JOW

## 입력 예제

ABC

BBBQ

BT

JOW

## 출력 결과

ABC

bbbq

bt

JOW

```
#include <iostream>
using namespace std;

const int sizeY = 4;
const int sizeX = 10;
const int offset = 'a' - 'A';    // 32

// ascii -> 'A' = 65, 'a' = 97 (gap = 32)

void makeSmall(char* str, int len)
{
    for (int i = 0; i < len; ++i)
        str[i] += offset;
}

int main(void)
{
    char input[sizeY][sizeX + 1] = {};
    for (int i = 0; i < sizeY; ++i)
        cin >> input[i];

    int lens[sizeY] = {};
    for (int k = 0; k < sizeY; ++k)
        lens[k] = strlen(input[k]);

    int maxIdx = 0;
    int minIdx = 0;
    for (int o = 0; o < sizeY; ++o)
    {
        if (lens[maxIdx] < lens[o])
            maxIdx = o;
        if (lens[minIdx] > lens[o])
            minIdx = o;
    }

    makeSmall(input[maxIdx], lens[maxIdx]);
    makeSmall(input[minIdx], lens[minIdx]);

    for (int m = 0; m < sizeY; ++m)
        cout << input[m] << endl;

    return 0;
}
```



## Level27 t\_magic 숫자 만들기

(난이도 : ★★☆☆)

문제 7번 [[숙제](#) [목록보기](#)]

3	5
1	

t\_magic 숫자란,

윗칸 + 왼쪽칸을 더해서 만든 숫자입니다

이곳에 들어갈 t\_magic 숫자는  $1+5 = 6$  입니다

1	2	1
4	①	②
2	③	④

이렇게 table이 있고

각 빈칸에 t\_magic 숫자로 채운다면 ①~④번 순서대로 값을 채우면 됩니다.

1	2	1
4	6	7
2	8	15

1			
1			

3x4 배열이 있고 맨 윗줄에 들어갈 숫자 4개를 입력 받아주세요. 그리고 나머지칸에 t\_magic 숫자로 채운 후 결과를 출력 해주세요.

Ex) 1 1 1 1 →

1	1	1	1
1	2	3	4
1	3	6	10

### 입력 예제

1 1 1 1

### 출력 결과

1 1 1 1

1 2 3 4

1 3 6 10

```

#include <iostream>
using namespace std;

const int sizeY = 3;
const int sizeX = 4;
int table[sizeY][sizeX] =
{
    0,0,0,0,
    1,0,0,0,
    1,0,0,0,
};

int tMagic(int _y, int _x)
{
    int result = 0;
    int a = table[_y - 1][_x];
    int b = table[_y][_x - 1];
    result = a + b;
    return result;
}

int main(void)
{
    for (int i = 0; i < sizeX; ++i)
        cin >> table[0][i];

    for (int y = 1; y < sizeY; ++y)
    {
        for (int x = 1; x < sizeX; ++x)
            table[y][x] = tMagic(y, x);
    }

    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
            cout << table[y][x] << " ";
        cout << endl;
    }

    return 0;
}

```

## Level27 2개의 max값 찾기 (난이도 : ★★☆☆)

문제 8번 [[숙제](#) [목록보기](#)]

3	5	1
9	2	7
6	11	3

9개의 숫자를 입력 받으세요.

그리고 **가장 큰 숫자**와 **두번째로 큰 숫자**를 찾아 값과 좌표를 출력 하세요.

ex)

[입력]

3 5 1

9 2 7

6 11 3

[출력]

첫번째:11(2,1)

두번째:9(1,0)

## 입력 예제

3 5 1

9 2 7

6 11 3

## 출력 결과

첫번째:11(2,1)

두번째:9(1,0)

```

#include <iostream>
using namespace std;

const int sizeY = 3;
const int sizeX = 3;
int table[sizeY][sizeX] = {};

struct Point
{
public:
    Point()
        : val(0)
        , x(0)
        , y(0)
    {
    }
    int val;
    int x;
    int y;
};

void findValue(Point* p)
{
    p->val = table[0][0];
    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
        {
            if (p->val < table[y][x])
            {
                p->val = table[y][x];
                p->y = y;
                p->x = x;
            }
        }
    }
}

void findValue(Point* p, int _max) // 오버로딩
{
    p->val = table[0][0];
    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
        {
            if (p->val < table[y][x] && table[y][x] < _max)
            {

```

```

        p->val = table[y][x];
        p->y = y;
        p->x = x;
    }
}

}

int main(void)
{
    for (int i = 0; i < sizeY * sizeX; ++i)
        cin >> table[i / sizeX][i % sizeX];

    Point first;
    Point second;
    findValue(&first);
    findValue(&second, first.val);

    cout << "첫번째:" << first.val << "(" << first.y << "," << first.x << ")\n";
    cout << "두번째:" << second.val << "(" << second.y << "," << second.x << ")\n";

    return 0;
}

```

# Level27 아기돼지 3형제

문제 9번 [숙제 [목록보기](#)]

4x4 배열의 땅이 있습니다.

#으로 표시되는 아기돼지 집 세곳을 입력 받으세요.

그리고 늑대에게 잡아먹히지 않도록 벽을 감싸주세요(8방향)

벽을 짓느라 아기돼지집을 부수면 안됩니다.

(빈공간은 언더바\_로 출력 해주세요)

Hint: Direct를 이용해주세요.

ex)

[입력]				
0 0				
2 2				
0 3				

→

#	@	@	#
@	@	@	@
	@	#	@
	@	@	@

[출력결과]

```
# @ @ #
@ @ @ @
  @ # @
  @ @ @
```

## 입력 예제

```
0 0
2 2
0 3
```

## 출력 결과

```
#@@#
@@@@
_@#@
_@@@
```

```

#include <iostream>
using namespace std;

const int sizeY = 4;
const int sizeX = 4;
char map[sizeY][sizeX] = {};
const int pigCnt = 3;

struct Point
{
    Point()
        : y(0)
        , x(0)
    {
    }
    int y;
    int x;
};

void drawPigPos(Point* p)
{
    for(int i = 0; i < pigCnt; ++i)
        map[p[i].y][p[i].x] = '#';
}

void drawWall(Point* p)
{
    const int directCnt = 8;
    const int axisCnt = 2;
    int Direct[directCnt][axisCnt] =
    {
        -1,0,    // top
        -1,1,
        0,1,     // right
        1,1,
        1,0,     // bottom
        1,-1,
        0,-1,    // left
        -1,-1
    };

    for (int i = 0; i < pigCnt; ++i)
    {
        int pigPosY = p[i].y;
        int pigPosX = p[i].x;
        for (int k = 0; k < directCnt; ++k)
        {

```

```

        int wallPosY = pigPosY + Direct[k][0];
        int wallPosX = pigPosX + Direct[k][1];
        if (wallPosY >= 0 && wallPosY < sizeY
            && wallPosX >= 0 && wallPosX < sizeX)
        {
            if (map[wallPosY][wallPosX] == '\0')
                map[wallPosY][wallPosX] = '@';
        }
    }
}

void printMap()
{
    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
        {
            if (map[y][x] == '\0')
                cout << '_';
            else
                cout << map[y][x];
        }
        cout << endl;
    }
}

int main(void)
{
    Point pigPos[pigCnt] = {};
    for (int i = 0; i < pigCnt; ++i)
        cin >> pigPos[i].y >> pigPos[i].x;

    drawPigPos(pigPos);
    drawWall(pigPos);
    printMap();

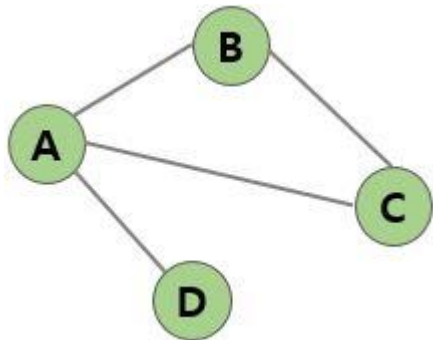
    return 0;
}

```



# Level27 가장 많이 연결된 알파벳 찾기

문제 10번 [숙제 목록보기]



A, B, C, D의 관계도가 그려져 있고 이것을 배열로 표현 하면

아래와 같이 표현 할 수 있습니다.

	A	B	C	D
A		1	1	1
B	1		1	
C	1	1		
D	1			

숫자 1은 연결됨을 나타냅니다.  
A와 C는 연결되어 있습니다.

위 배열에서 가장 많이 연결된 알파벳은 A 입니다.

위와 같이 4x4 배열에 연결상태를 입력받고, 가장 많이 연결된 알파벳을 출력 하세요.

## 입력 예제

```
0 1 1 1
1 0 1 0
1 1 0 0
1 0 0 0
```

## 출력 결과

A

```
#include <iostream>
using namespace std;

const int graphSize = 4;
int graph[graphSize][graphSize] = {};
const int offset = 65; // ascii-> 'A' = 65

int main(void)
{
    for (int i = 0; i < graphSize * graphSize; ++i)
        cin >> graph[i / graphSize][i % graphSize];

    int lines[graphSize] = {};
    for (int y = 0; y < graphSize; ++y)
    {
        for (int x = 0; x < graphSize; ++x)
        {
            lines[y] += graph[y][x];
        }
    }

    int maxIdx = 0;
    for (int k = 0; k < graphSize; ++k)
    {
        if (lines[maxIdx] < lines[k])
            maxIdx = k;
    }

    char result = maxIdx + offset;
    cout << result;

    return 0;
}
```

## Level27 꽃이 피었습니다. (난이도 : ★★★)

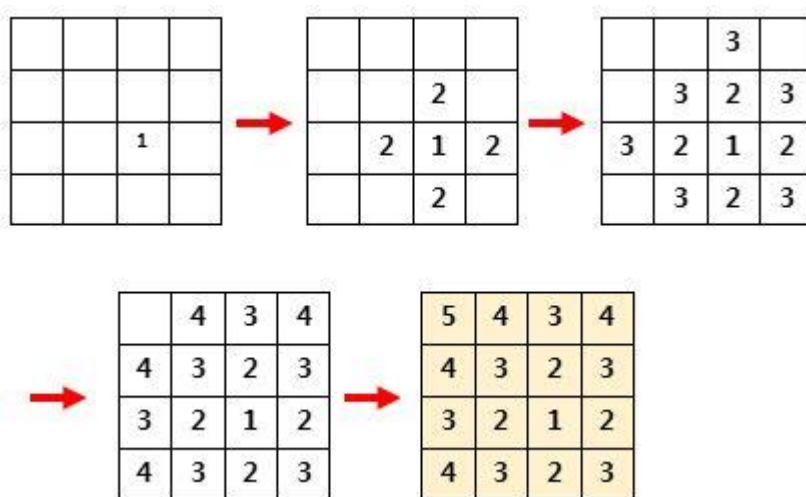
문제 11번 [[숙제](#) [목록보기](#)]

숫자 1의 위치를 입력 받으세요. (y, x)

그리고 1의 위치를 찾아내서 위, 아래, 왼쪽, 오른쪽칸의 빈칸에 숫자 2를 채워주세요.

그리고 숫자 2를 찾아서 모든 숫자 2에서 위, 아래, 오른쪽, 아래칸의 빈칸에 숫자 3을 채워주세요.

이 과정을 반복해서 배열을 아래와 같이 배열을 꽉 채운 후 결과를 출력 해주세요.



## 입력 예제

2 2

## 출력 결과

```
5 4 3 4
4 3 2 3
3 2 1 2
4 3 2 3
```

```

#include <iostream>
using namespace std;

const int sizeY = 4;
const int sizeX = 4;
int map[sizeY][sizeX] = {};
const int directCnt = 4;
const int axisCnt = 2;
int direct[directCnt][axisCnt] =
{
    -1,0,    // top
    0,1,     // right
    1,0,     // bottom
    0,-1    // left
};

void placeValue(int _y, int _x, int _val)
{
    map[_y][_x] = _val;
}

void func2(int _y, int _x, int _val)
// 해당 좌표를 기준으로 상하좌우에 '기준값+1'을 넣어준다.
{
    int val = _val + 1;
    for (int i = 0; i < directCnt; ++i)
    {
        int y = _y + direct[i][0];
        int x = _x + direct[i][1];
        if (y >= 0 && y < sizeY && x >= 0 && x < sizeX)
        {
            if (map[y][x] == 0)
                placeValue(y, x, val);
        }
    }
}

void func1(int _val)
// 기준값에 해당하는 값을 가진 좌표를 찾아준다.
{
    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
        {
            if (map[y][x] == _val)
            {
                func2(y, x, _val);
            }
        }
    }
}

```

```

    }
}

}

bool isFull()    // 가득차면 true, 빈 곳이 있으면 false
{
    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
        {
            if (map[y][x] == 0)
                return false;
        }
    }

    return true;
}

void printMap()
{
    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
            cout << map[y][x] << " ";
        cout << endl;
    }
}

int main(void)
{
    int inputY = 0;
    int inputX = 0;
    const int startVal = 1;
    cin >> inputY >> inputX;
    placeValue(inputY, inputX, startVal);

    int val = startVal;
    while (true)
    {
        func1(val);
        if (isFull())
            break;
        ++val;
    }

    printMap();
}

```

```
}    return 0;
```

## Level27 삼각관계

문제 12번 [[숙제 목록보기](#)]

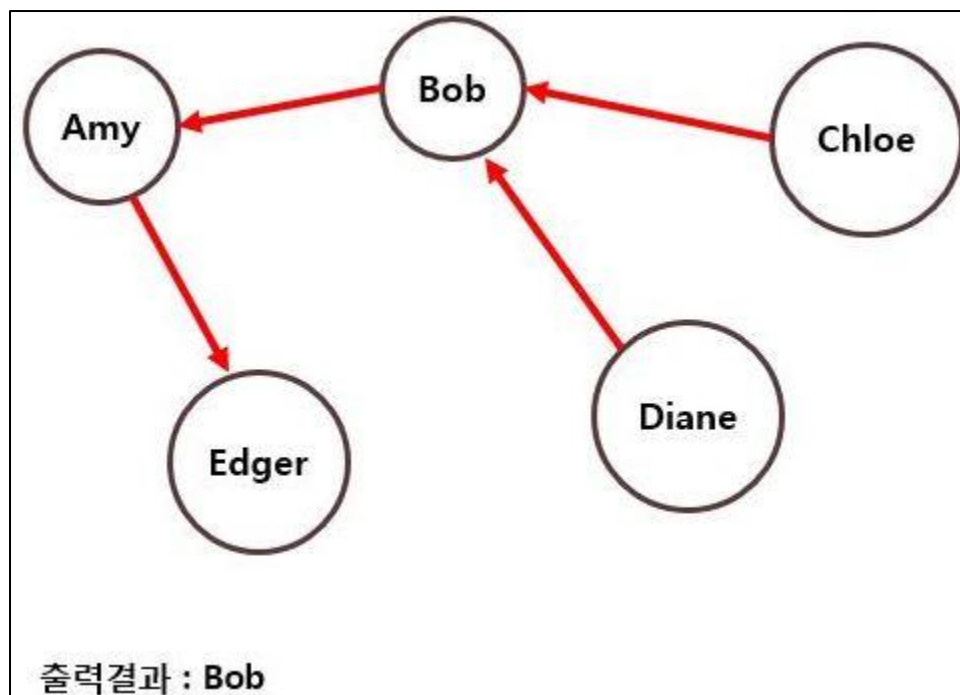
드라마의 삼각관계를 나타낸 그래프입니다.

그래프를 인접행렬로 작성 해 주세요.

인접행렬을 이용해서

다섯명의 사람 중 가장 인기가 좋은 사람을 찾아서 출력 해주세요.

(A -----> B는 A가 B를 좋아한다는 뜻입니다.)



## 출력 결과

Bob

```

#include <iostream>
using namespace std;

const int sizeY = 5;
const int sizeX = 5;
const int people = 5;
const int length = 10;

int main(void)
{
    /*
        Army    0    0    0    0    1
        Bob      1    0    0    0    0
        Chole    0    1    0    0    0
        Diane    0    1    0    0    0
        Edger    0    0    0    0    0

    */
    char names[people][length] =
    {
        "Army",
        "Bob",
        "Chole",
        "Diane",
        "Edger",
    };

    int visiter[people] = {};

    bool table[sizeY][sizeX] =
    {
        0,0,0,0,1,
        1,0,0,0,0,
        0,1,0,0,0,
        0,1,0,0,0,
        0,0,0,0,0,
    };

    for (int x = 0; x < sizeX; ++x)
    {
        int idx = x;
        for (int y = 0; y < sizeY; ++y)
        {
            if (table[y][x])
                ++visiter[idx];
        }
    }
}

```



```
int maxIdx = 0;
for (int k = 0; k < people; ++k)
{
    if (visiter[maxIdx] < visiter[k])
        maxIdx = k;
}

cout << names[maxIdx];

return 0;
}
```

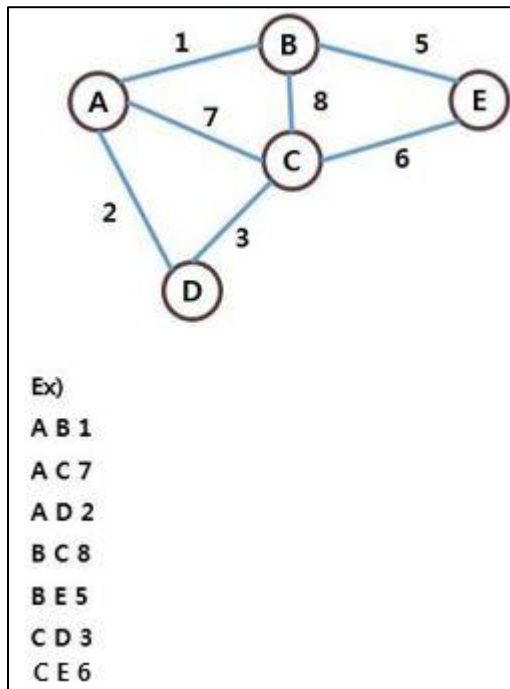
# Level27 인접행렬 그래프

문제 13번 [[숙제](#) [목록보기](#)]

아래와 같이 5개 노드를 가진 그래프를 인접행렬로 하드코딩 해 주세요

그리고 어떤 간선들이 존재하는지 아래와 같이 출력 해주세요.

(입력값은 없습니다)



## 출력 결과

A B 1

A C 7

A D 2

B C 8

B E 5

C D 3

C E 6

```

#include <iostream>
using namespace std;

const int sizeY = 5;
const int sizeX = 5;
const int people = 5;

int main(void)
{
    char names[people] = { 'A', 'B', 'C', 'D', 'E' };
    int graph[sizeY][sizeX] =
    {
        //      A B C D E
        0,1,7,2,0,    // A
        0,0,8,0,5,    // B
        0,0,0,3,6,    // C
        0,0,0,0,0,    // D
        0,0,0,0,0,    // E
    };

    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
        {
            if (graph[y][x] != 0)
                cout << names[y] << " "
                    << names[x] << " "
                    << graph[y][x] << endl;
        }
    }

    return 0;
}

```

# Level27 링크드리스트 라인

문제 14번 [[숙제](#) [목록보기](#)]

링크드리스트로 라인을 구현하고자 합니다.

입력받은 문자들을 모두 링크드리스트에 등록을 하고

등록이 끝난 후

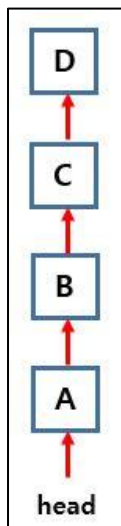
다시 처음부터 출력을 해 주세요

\* add 함수를 만들어서 작성 해 주세요

ex)

[입력]

4  
A B C D



## 입력 예제

4

A B C D

## 출력 결과

A B C D

```

#include <iostream>
#define _CRTDBG_MAP_ALLOC
#include <stdlib.h>
#include <crtDBG.h>
using namespace std;

class Line
{
public:
    Line()
        : head(nullptr)
        , tail(nullptr)
    {

    }

    struct Node
    {
    public:
        Node()
            : val('\0')
            , pNext(nullptr)
        {

        }
        Node(char data)
            : val(data)
            , pNext(nullptr)
        {

        }
        char val;
        Node* pNext;
    };

    void addNode(char _data)
    {
        if (head == nullptr)
        {
            head = new Node(_data);
            tail = head;
        }
        else
        {
            tail->pNext = new Node(_data);
            tail = tail->pNext;
        }
    }
}

```

```

void print()
{
    Node* temp = head;
    while (true)
    {
        if (temp == nullptr)
            break;
        cout << temp->val << " ";
        temp = temp->pNext;
    }
}

void delNode()
{
    Node* temp1 = head;
    Node* temp2 = nullptr;
    while (true)
    {
        if (temp1 == head)
            head == nullptr;
        temp2 = temp1->pNext;
        delete temp1;
        if (temp2 == tail)
        {
            tail == nullptr;
            delete temp2;
            break;
        }
        else
            temp1 = temp2;
    }
}

Node* head;
Node* tail;
};

int main(void)
{
    int size = 0;
    cin >> size;
    char* vals = new char[size + 1];
    for (int i = 0; i < size; ++i)
        cin >> vals[i];

    Line* line = new Line();

```

```
    for(int k = 0; k < size; ++k)
        line->addNode(vals[k]);
    line->print();

    line->delNode();
    delete line;
    delete[] vals;

    _CrtDumpMemoryLeaks();
    return 0;
}
```

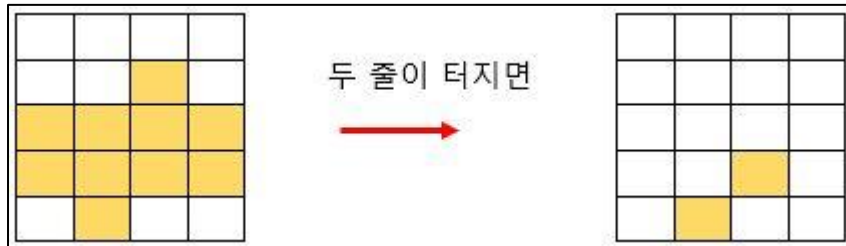
# Level27 팍찬줄 터트리기

문제 15번 [숙제 목록보기]

테트리스게임에서는 한줄이 팍 찰때 그 줄은 사라지게 됩니다.

블럭 상태를 입력 받고, 팍찬줄이 터트려지고 난 후의 결과를 출력 해주세요.

예시)



\* 터진 줄만 제거하면 됩니다. 위에있는 블럭이 완전히 바닥으로 떨어지는 것이 아닙니다.

## 입력 예제

```
0 0 0 0
0 0 1 0
1 1 1 1
1 1 1 1
0 1 0 0
```

## 출력 결과

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 1 0
0 1 0 0
```



```

#include <iostream>
using namespace std;

const int sizeY = 5;
const int sizeX = 4;
int stage[sizeY][sizeX] = {};

void input()
{
    for (int y = 0; y < sizeY; ++y)
    {
        for (int x = 0; x < sizeX; ++x)
            cin >> stage[y][x];
    }
}

void bingo(int _y)
{
    for (int x = 0; x < sizeX; ++x)
        stage[_y][x] = 0;
}

void modify(int _y)
{
    for (int y = _y - 1; y >= 0; --y)
    {
        for (int x = 0; x < sizeX; ++x)
        {
            stage[y + 1][x] = stage[y][x];
        }
    }
}

void check()
{
    for (int y = 0; y < sizeY; ++y)
    {
        int cnt = 0;
        for (int x = 0; x < sizeX; ++x)
        {
            if (stage[y][x] != 0)
                ++cnt;
        }
        if (cnt == sizeX)
        {
            bingo(y);
            modify(y);
        }
    }
}

```

```
        }  
    }  
}  
  
void print()  
{  
    for (int y = 0; y < sizeY; ++y)  
    {  
        for (int x = 0; x < sizeX; ++x)  
            cout << stage[y][x];  
        cout << endl;  
    }  
}  
  
int main(void)  
{  
    input();  
    check();  
    print();  
  
    return 0;  
}
```