(훈련반1) Level21.5

중력문제 등 설계를 해야만 풀 수 있는 문제들이 나옵니다.

설계후 코딩을 하고, 버그가 발생하면 어디서 버그가 발생하는지

최대한 찾아보세요! 고민하고 원인을 찾는 시간만큼 코딩 실력이 늘게 됩니다.

(트레이스를 아무리 해도 디버깅이 잘 안되면 언제든 질문주세요)

Level21.5 너에게 가려면 [난이도 : 3]

문제 1번 [숙제 목록보기]

4 x 3 배열에 A, B가 하나씩 적혀 있습니다.

A와 B를 찾아 상하좌우로 몇칸 떨어져 있는지 출력하면 됩니다.

A위치에서 오른쪽 한칸, 밑으로 두칸 움직이면 됩니다.

답은 총 **세칸** 입니다.

입력 예제

```
DTK
EAP
CQG
PHB
```

출력 결과

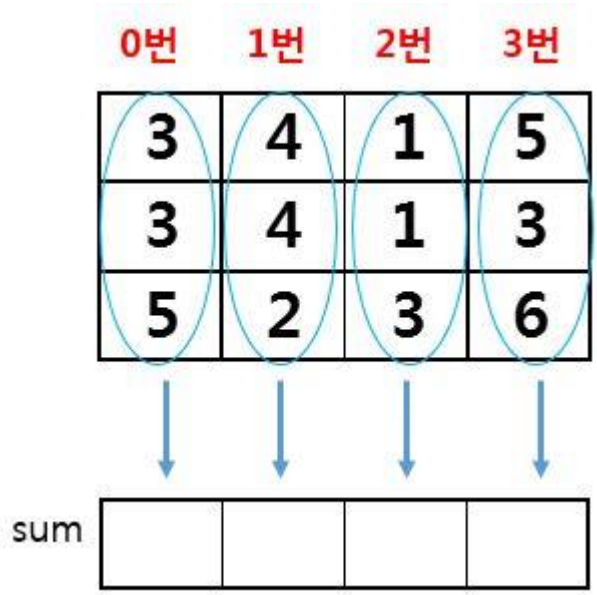
```
3
```

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

class Point
{
public:
	Point(int y, int x, char val)
		: x(x)
		, y(y)
		, val(val)
	{

	}
	int x;
	int y;
	char val;
};

int signCheck(int num)
{
	if (num < 0)
		num *= (-1);
	return num;
}

const int lenY = 4;
const int lenX = 3;
const int allFind = 2;

int main(void)
{
	char map[lenY][lenX] = {};
	for (int y = 0; y < lenY; ++y)
	{
		for (int x = 0; x < lenX; ++x)
			cin >> map[y][x];
	}

	Point a(0, 0, 'A');
	Point b(0, 0, 'B');
	int changeCnt = 0;

	for (int y = 0; y < lenY; ++y)
	{
		for (int x = 0; x < lenX; ++x)
		{
			if (map[y][x] == a.val)
```

```
                        {
                                a.y = y;
                                a.x = x;
                                changeCnt++;
                        }

                        if (map[y][x] ==b.val)
                        {
                                b.y = y;
                                b.x = x;
                                changeCnt++;
                        }
                        if (changeCnt == allFind)
                                break;
                }
                if (changeCnt == allFind)
                        break;
        }

        int gapX = signCheck(a.x - b.x);
        int gapY = signCheck(a.y - b.y);
        int gapSum = gapX + gapY;

        cout << gapSum;

        return 0;
}
```

Level21.5 세로줄의 합과 해당 인덱스값 구하기 [난이도 : 2]

문제 2번 [숙제 목록보기]



각 세로줄의 합을 구해서 sum이라는 배열에 넣어주세요.

그리고 index라는 변수에 숫자 하나를 입력받고,

sum[index] 값을 출력 해주세요.

입력 예제

```
2
```

출력 결과

```
5
```

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

const int mapY = 3;
const int mapX = 4;
const int sumLen = 4;

int main(void)
{
	int map[mapY][mapX] =
	{
		3,4,1,5,
		3,4,1,3,
		5,2,3,6
	};

	int sum[sumLen] = {};
	int sumIdx = 0;
	for (int x = 0; x < 4; ++x)
	{
		for (int y = 0; y < 3; ++y)
		{
			sum[sumIdx] += map[y][x];
		}
		++sumIdx;
	}

	int index = 0;
	cin >> index;

	cout << sum[index];

	return 0;
}
```

Level21.5 문자 양옆으로 #넣기

[난이도 : 5]

문제 3번 [숙제 목록보기]

문장 하나와 문자 2개를 입력받아주세요

문장에서, 문자가 존재하는 곳 좌우의 값을 '#'으로 바꾸어 출력 해 주세요

* 입력받은 문자는 문장에 각각 1개씩만 존재합니다.

* 예제3와 같이, #을 넣는게 불가능하다면 넣지 않습니다.

* 설계를 꼼꼼히 하신 후 풀어야 합니다.

예제1) **APKDB**를 입력받고 **P D**를 입력받으면 **#P#D#**을 출력하면 됩니다.

예제2) **REWUQ**를 입력받고 **W U**를 입력받으면 **R####**이 출력되면 됩니다

예제3) **ABCDEFG**를 입력받고, **A G**를 입력받으면 **A#CDE#G**를 출력해야 합니다.

입력 예제

```
APKDB
P D
```

출력 결과

```
#P#D#
```

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

class Position
{
public:
	Position()
		: idx(0)
		, val('\0')
	{

	}
	int idx;
	char val;
};

const int inputCnt = 2;
const int maxStrLength = 10;

int main(void)
{
	char* str = new char[maxStrLength];
	Position arr[inputCnt] = {};
	cin >> str;
	cin >> arr[0].val >> arr[1].val;

	int length = strlen(str);// char* 을 인자값으로 넣어준다.
	for (int i = 0; i < length; ++i)
	{
		if (str[i] == arr[0].val)
			arr[0].idx = i;
		if (str[i] == arr[1].val)
			arr[1].idx = i;
	}

	int left = 0;
	int right = 0;
	for (int k = 0; k < inputCnt; ++k)
	{
		left = arr[k].idx - 1;
		right = arr[k].idx + 1;
		if (left >= 0 && left < length)
			str[left] = '#';
		if (right >= 0 && right < length)
			str[right] = '#';
	}
```

```
        cout << str;

        delete[] str;
        return 0;
}
```

Level21.5 중력문제 [난이도 : 6]

문제 4번 [숙제 목록보기]

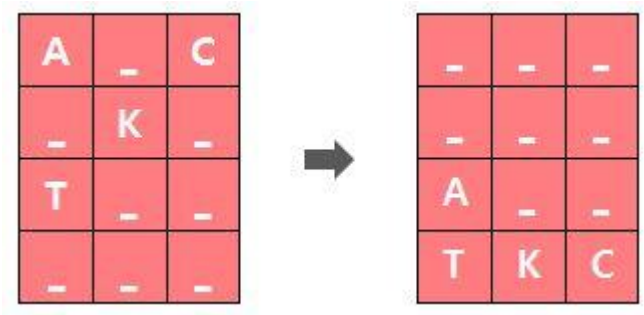
우주선인 2차배열에 문자들이 둥둥 공중에 떠있습니다.

소행성과의 충돌 위험으로 중력이 있는 행성에 불시착하였더니

문자들이 바닥에 떨어졌습니다.

아래 그림처럼 입력하고 중력을 받아 바닥에 떨어진 문자상태를 출력하면 됩니다.

[입력] **[출력]**



입력 예제

```
A_C
_K_
T__
___
```

출력 결과

```
___
___
A__
TKC
```

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

const int arrY = 4;
const int arrX = 3;

void swap(char* ch1, char* ch2)
{
	char temp = *ch1;
	*ch1 = *ch2;
	*ch2 = temp;
}

// 아래쪽 이동 공간이 있는지 확인
bool isEmpty(char (*arr)[arrX], int y, int x)
{
	int bottom = y + 1;
	if (bottom >= 0 && bottom < arrY)
	{
		if (arr[bottom][x] == '_')
		{
			return true;	// 내려갈 공간이 있음
		}
	}
	return false;			// 내려갈 공간이 없음
}

int main(void)
{
	char starship[arrY][arrX] = {};
	for (int y = 0; y < arrY; ++y)
	{
		for (int x = 0; x < arrX; ++x)
		{
			cin >> starship[y][x];
		}
	}
	int air = arrY - 2;
	for (int y = air; y >= 0 ; --y)
	{
		for (int x = 0; x < arrX; ++x)
		{
			if (starship[y][x] != '_')	// 해당좌표 (y,x)에 알파벳이 있음?
			{
				int targetY = y;
				while (isEmpty(starship, targetY, x))	// 아래 내려갈
공간이 있음?
```

```
                                {
                                        int bottom = targetY + 1;
                                        if (bottom < arrY)
                                        {
                                                swap(starship[targetY][x],
starship[bottom][x]);
                                                ++targetY;
                                        }
                                }
                        }
                }
        }

        for (int y = 0; y < arrY; ++y)
        {
                for (int x = 0; x < arrX; ++x)
                        cout << starship[y][x];
                cout << endl;
        }

        return 0;
}
```

Level21.5 counting 후 정렬하기

[난이도 : 4]

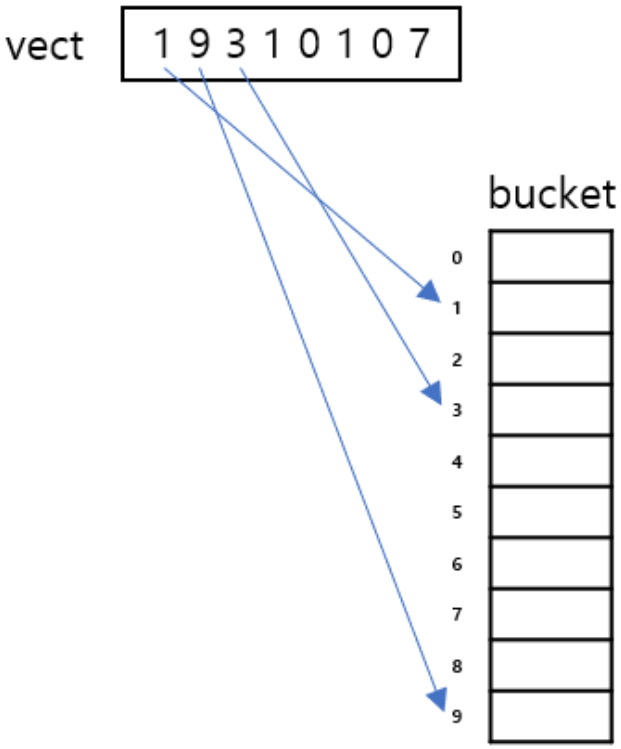
문제 5번 [숙제 목록보기]

숫자 8개(0 ~ 9)를 입력받습니다.

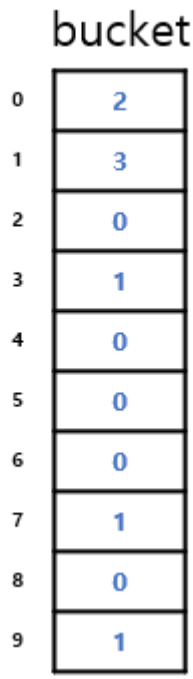
입력받은 숫자들을 아래의 규칙을 이용해서 정렬 후 출력 해 주세요.

[규칙]

먼저, 입력받은 vect배열 값들을 하나씩 탐색하면서 bucket 배열에 counting 합니다.



이렇게 counting 후 만들어지는 배열은 다음과 같습니다.



이제 bucket 배열을 이용해서 정렬된 결과를 출력하면 됩니다.

위 예제에서

숫자 0은 2회 발견되었으니 '0' 2회 출력

숫자 1은 3회 발견되었으니 '1' 3회 출력

숫자 3은 1회 발견되었으니 '3' 1회 출력

...

숫자 9는 1회 발견되었으니 '9' 1회 출력

따라서 출력되는 결과는 다음과 같습니다.

0 0 1 1 1 3 7 9

입력 예제

```
1 9 3 1 0 1 0 7
```

출력 결과

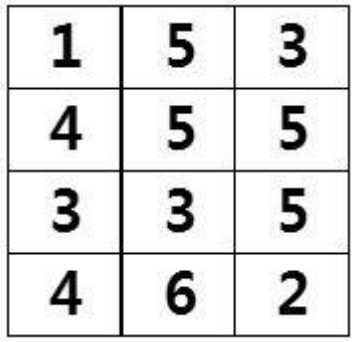
```
0 0 1 1 1 3 7 9
```

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

const int vectLen = 8;
const int bucketLen = 10;

int main(void)
{
	int vect[vectLen] = {};
	for (int i = 0; i < vectLen; ++i)
		cin >> vect[i];

	int bucket[bucketLen] = {};
	for (int k = 0; k < vectLen; ++k)
	{
		int targetIdx = vect[k];
		bucket[targetIdx]++;
	}

	for (int m = 0; m < bucketLen; ++m)
	{
		int targetCnt = bucket[m];
		if (targetCnt != 0)
		{
			for (int n = 0; n < targetCnt; ++n)
				cout << m << " ";
		}
	}

	return 0;
}
```

Level21.5 범위의 숫자 #으로 바꾸기 [난이도 : 2]

문제 6번 [숙제 목록보기]

아래 배열을 하드코딩 해 주세요



1	5	3
4	5	5
3	3	5
4	6	2

숫자 두개를 변수 a, b에 입력 받으세요.

a ~ b 사이에 있는 숫자들을 모두 0으로 바꿔 주세요. (a <= x <= b)

그리고 출력할때 **0인 부분만 #으로 바꿔 출력해주세요.**

입력 예제

```
3 4
```

출력 결과

```
1 5 #
# 5 5
# # 5
# 6 2
```

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

const int arrY = 4;
const int arrX = 4;

char a = '\0';
char b = '\0';

void isContain(char* ch)
{
	if (*ch >= a && *ch <= b)
	{
		*ch = '#';
	}
}

int main(void)
{
	char arr2D[arrY][arrX] =
	{
		"153",
		"455",
		"335",
		"462"
	};
	// ascii -> '0' = 48, '9' = 57
	cin >> a >> b;
	if (a > b)
	{
		char temp = a;
		a = b;
		b = temp;
	}

	for (int y = 0; y < arrY; ++y)
	{
		for (int x = 0; x < (arrX - 1); ++x)
			isContain(&arr2D[y][x]);
	}

	return 0;
}
```

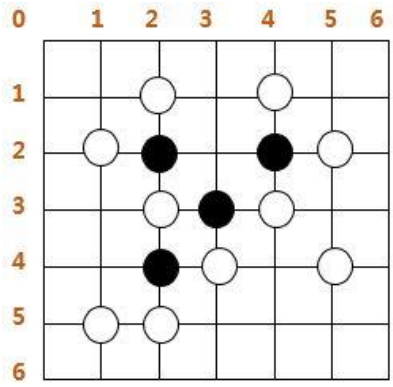
Level21.5 바둑이 게임 [난이도 : 5]

문제 7번 [숙제 목록보기]

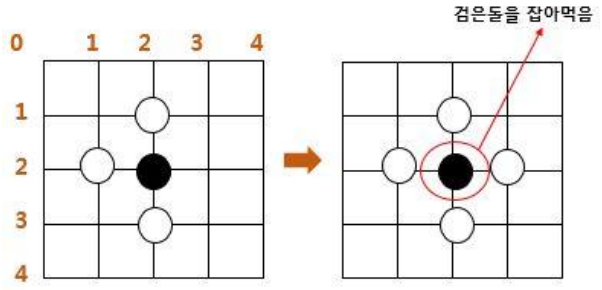
바둑이판 상태를 하드코딩 해주세요.

흰돌을 놓을 곳(좌표)을 입력 받으세요.

입력받은 좌표에 흰돌을 놓을때, 돌 몇개를 잡아먹을 수 있는지 출력 하세요.



위, 아래, 왼쪽, 오른쪽을 감싸고 있을때만 안쪽에 있는 돌을 잡아 먹을 수 있습니다.



이 상태에서 (2,3)에 흰돌을 놓으면 검은돌을 잡아 먹을 수 있습니다.

*** 이 게임은 바둑과는 다릅니다.**

검을돌을 잡아먹을 수 있는 방법은 오로지 위, 아래, 왼쪽, 오른쪽을 감쌀때만 입니다.

*** direct를 써서 풀어주세요**

입력 예제

```
2 3
```

출력 결과

```
3
```

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

const int stageY = 7;
const int stageX = 8;
const int die = 4;

char omok[stageY][stageX] =
{
        "NNNNNNN",
        "NNWNWNN",
        "NWBNBWN",
        "NNWBWNN",
        "NNBWNWN",
        "NWWNNNN",
        "NNNNNNN"
};

bool isDie(int y, int x)
{
        int offset[4][2] =
        {
                -1, 0,  // top
                0, 1,   // right
                1, 0,   // bottom
                0, -1   // left
        };

        int whiteCnt = 0;
        for (int i = 0; i < 4; ++i)
        {
                int chY = y + offset[i][0];
                int chX = x + offset[i][1];
                if (omok[chY][chX] != 'W')
                        break;
                else
                        ++whiteCnt;
        }

        if(whiteCnt == die)
                return true;
        return false;
}

int main(void)
{
        int newX = 0, newY = 0;
```

```
        cin >> newY >> newX;
        omok[newY][newX] = 'W';

        int score = 0;
        for (int y = 0; y < stageY; ++y)
        {
                for (int x = 0; x < stageX - 1; ++x)
                {
                        if (omok[y][x] == 'B')
                        {
                                if (isDie(y, x))
                                        ++score;
                        }
                }
        }

        cout << score;

        return 0;
}
```

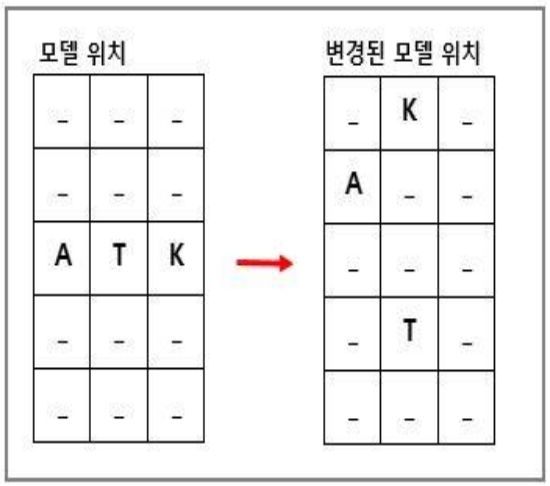
Level21.5 모델 위치 지시하기

[난이도 : 5]

문제 8번 [숙제 목록보기]

모델들이 한 줄로 무대위에 서있습니다. 디렉터는 모델들에게 **지시를 7번** 합니다.

지시 이후 모델들의 위치를 출력 해주세요.**(굉장한 노가다 문제입니다)**



위치 지령

A UP

T DOWN

K UP

A RIGHT

K UP

K LEFT

A LEFT

입력 예제

```
A UP
T DOWN
K UP
A RIGHT
K UP
K LEFT
A LEFT
```

출력 결과

```
_K_
A__
___
_T_
```

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

const int cmdCount = 7;
const int cmdMaxLength = 6;
const int stageY = 5;
const int stageX = 3;

class Model
{
public:
	Model(char n, int y, int x)
		: name(n)
		, y(y)
		, x(x)
	{

	}
	char name;
	int x;
	int y;
};

void move(Model* model, char* cmd)
{
	if (strcmp(cmd, "UP") == 0)
		--(model->y);
	else if (strcmp(cmd, "DOWN") == 0)
		++(model->y);
	else if (strcmp(cmd, "RIGHT") == 0)
		++(model->x);
	else if (strcmp(cmd, "LEFT") == 0)
		--(model->x);
	else
		__noop;
}

int main(void)
{
	char stage[stageY][stageX + 1] =
	{
		"___",
		"___",
		"___",
		"___",
		"___"
	};
```

```
        Model A('A', 2, 0);
        Model T('T', 2, 1);
        Model K('K', 2, 2);

        char model[cmdCount] = {};        //  7번
        char command[cmdCount][cmdMaxLength] = {};
        for (int i = 0; i < cmdCount; ++i)
        {
                cin >> model[i] >> command[i];
        }

        for (int k = 0; k < cmdCount; ++k)
        {
                if (model[k] == 'A')
                        move(&A, command[k]);
                else if(model[k] == 'T')
                        move(&T, command[k]);
                else if (model[k] == 'K')
                        move(&K, command[k]);
        }

        stage[A.y][A.x] = A.name;
        stage[T.y][T.x] = T.name;
        stage[K.y][K.x] = K.name;

        for (int y = 0; y < stageY; ++y)
        {
                for (int x = 0; x < stageX; ++x)
                        cout << stage[y][x];
                cout << endl;
        }

        return 0;
}
```