

알고리즘 - 코딩 과제 #2

수요일반, 02분반, 정보보호학과, 2020111340, 우은지

(1) - 1. 문제정의

1	3	6	10	15
2	5	9	14	19
4	8	13	18	22
7	12	17	21	24
11	16	20	23	25

<그림 1>

다음과 같이 오른쪽 대각선 위로 칸이 없을 때까지 올라가는 형태로 배열을 채워야 한다.

채워지는 순서대로 인덱스 번호를 생각해 보면

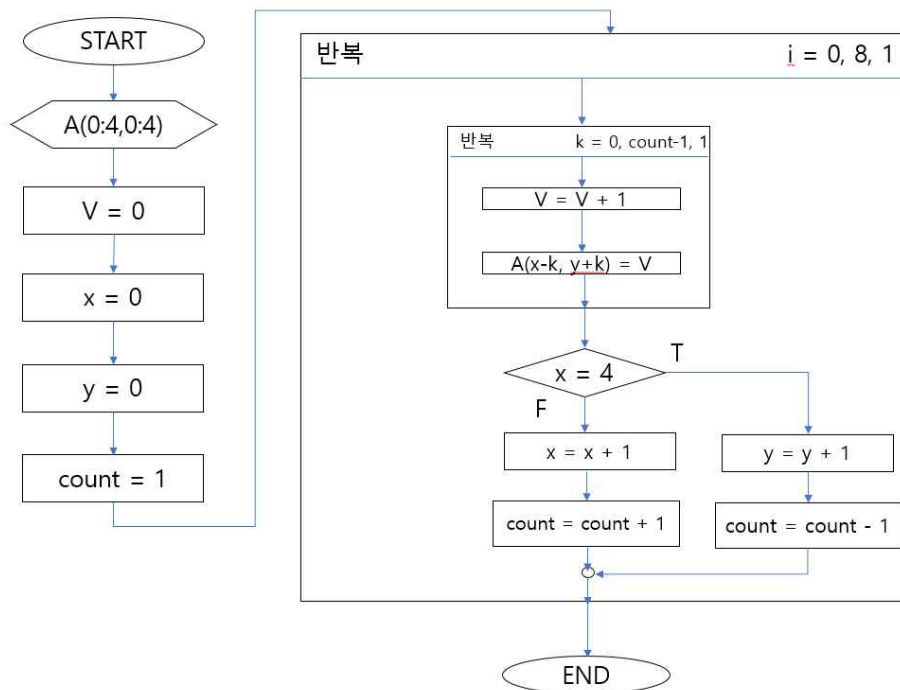
(0,0), (1,0), (0,1), (2,0), (1,1), (0,2), (3,0), (2,1),..., (4,3), (3,4), (4,4)와 같이 채워진다.

특징을 생각해 보면 대각선 형태로 9번 실행되는 것을 알 수 있고,

1개의 대각선이 채우는 칸의 개수를 순서대로 나열해 보면 1, 2, 3, 4, 5, 4, 3, 2, 1개의 칸을 채운다는 것을 알 수 있다.

1. 따라서 count 변수가 (칸을 몇 칸 채울지 새어주는 변수) 5까진 증가하고 5 다음은 1씩 감소하도록 구현해야 한다.
2. 오른쪽 위를 향한 대각선 형태로 올라가기 때문에 행 인덱스는 -1씩 감소되고 열 인덱스는 +1씩 되도록 구현해야 한다.
3. count가 5까진 행 인덱스를 1씩 증가시켜줘야 하지만 5가 넘어가는 순간 행 인덱스가 아닌 열 인덱스를 증가시켜줘야 한다.

(1) - 2. 설계 (순서도와 설명)



x: 배열의 행을 나타낸다.

y: 배열의 열을 나타낸다.

count: 한 번의 대각선 방향에서 몇 칸을 채울지 나타낸다.

- 배열의 인덱스를 c언어와 같게 하기 위해서 0년부터 4번 인덱스를 사용하여 배열을 표현한다.
- 대각선 한 번에 1, 2, 3, 4, 5, 4, 3, 2, 1 만큼씩 반복되도록 하기 위해 count 변수를 이용한다.
- 오른쪽 위로 올라가는 대각선 모양이니까 x는 1씩 감소, y는 1씩 증가시킨다.
- 5번째 대각선까진 x의 값 1씩 증가, count 값 1씩 증가시킨다.
- 6번째 대각선부터 x의 값은 그대로 두고 y의 값 1씩 증가, count 값 1씩 감소시킨다.

(1) - 3. 구현 (프로그램 소스)

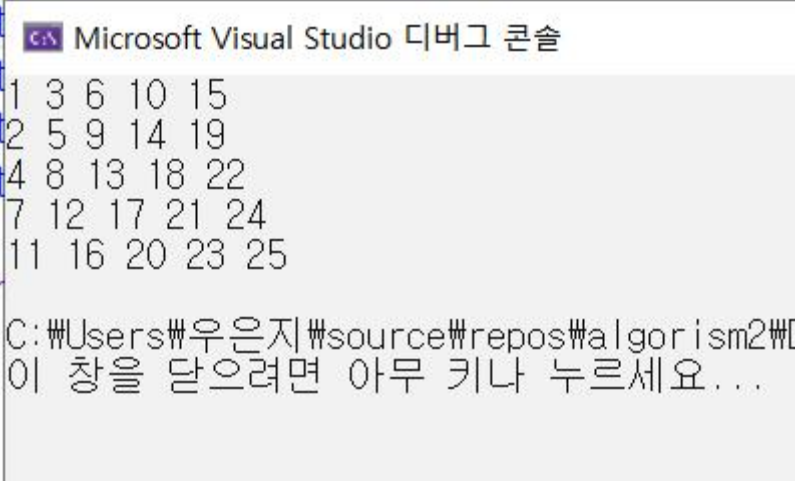
```
#include <stdio.h>

void main() {
    int A[5][5] = { 0 };
    int V = 0;
    int x = 0, y = 0;
    int count = 1;

    for (int i = 0; i < 9; i++) {
        for (int k = 0; k < count; k++) {
            V = V + 1;
            A[x-k][y + k] = V; // 오른쪽 위로 올라가는 대각선이니까
        }
        if (x == 4) { // 5번째 대각선을 수행하고부터는 x의 값을 건들지 않으니까 계
            속 해당 if문으로 오게 됨
            y = y + 1;
            count = count - 1;
        }
        else { // 1,2 3,4번째 대각선을 수행하고 여기로 옴
            x = x + 1;
            count = count + 1;
        }
    }

    for (int i = 0; i < 5; i++) { // 배열 출력
        for (int j = 0; j < 5; j++) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }
}
```

(1) - 4. 실행화면



```
Microsoft Visual Studio 디버그 콘솔  
1 3 6 10 15  
2 5 9 14 19  
4 8 13 18 22  
7 12 17 21 24  
11 16 20 23 25  
  
C:\Users\우은지\source\repos\algorithm2\algorithm2\bin\Debug\algorithm2.exe  
이 창을 닫으려면 아무 키나 누르세요...
```

(2) - 1. 문제정의

1	3	4	10	11
2	5	9	12	19
6	8	13	18	20
7	14	17	21	24
15	16	22	23	25

<그림 2>

다음과 같이 왼쪽 아래로 대각선으로 내려왔다가 오른쪽 위 대각선으로 올라가는 게 9번 반복된다.

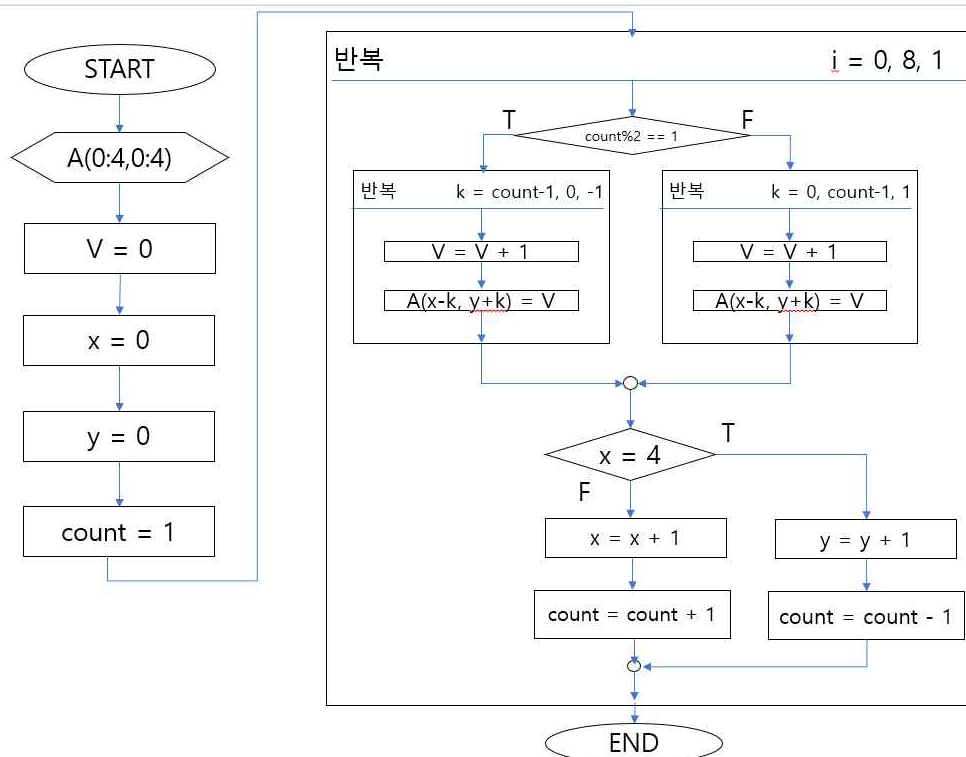
(0,0), (1,0), (0,1), (0,2), (1,1), (2,0), (3,0), (2,1),..., (4,3), (3,4), (4,4)와 같이 채워진다.

특징을 생각해 보면 왼쪽 아래 대각선, 오른쪽 위 대각선 모양으로 칸이 번갈아가며 채워지고,

대각선이 채우는 칸의 개수를 순서대로 나열해 보면 1, 2, 3, 4, 5, 4, 3, 2, 1 개의 칸을 채운다는 것을 알 수 있다.

1. 따라서 count 변수가 (칸을 몇 칸 채울지 새어주는 변수) 5까진 증가하고 5 다음은 1씩 감소하도록 구현해야 한다.
2. 왼쪽 아래를 향한 대각선 형태로 올라가는 경우엔 행 인덱스는 1씩 증가되고 열 인덱스는 -1씩 되도록 구현해야 한다.
3. 오른쪽 위를 향한 대각선 형태로 올라가는 경우엔 행 인덱스는 -1씩 감소되고 열 인덱스는 +1씩 되도록 구현해야 함
4. 2번과 3번이 반복되도록 해야 한다.
5. count가 5까진 행 인덱스를 1씩 증가시켜줘야 하지만 5가 넘어가는 순간 행 인덱스가 아닌 열 인덱스를 증가시켜줘야 한다.

(2) - 2. 설계 (순서도와 설명)



x: 배열의 행을 나타낸다.

y: 배열의 열을 나타낸다.

count: 한 번의 대각선 방향에서 몇 칸을 채울지 나타낸다.

- 배열의 인덱스를 c언어와 같게 하기 위해서 0년부터 4번 인덱스를 사용하여 배열을 표현한다.
- 대각선 한 번에 1, 2, 3, 4, 5, 4, 3, 2, 1 만큼씩 반복되도록 하기 위해 count 변수를 이용한다.
- count가 홀수인 경우, 왼쪽 아래로 내려가는 대각선 모양이니까 x는 1씩 증가, y는 1씩 감소시킨다.
- count가 짝수인 경우, 오른쪽 위로 올라가는 대각선 모양이니까 x는 1씩 감소, y는 1씩 증가시킨다.
- 5번째 대각선까진 x의 값 1씩 증가, count 값 1씩 증가시킨다.
- 6번째 대각선부터 x의 값은 그대로 두고 y의 값 1씩 증가, count 값 1씩 감소시킨다.

(2) - 3. 구현 (프로그램 소스)

```
#include <stdio.h>
```

```
void main() {
    int A[5][5] = { 0 };
    int V = 0;
    int x = 0, y = 0;
    int count = 1;

    for (int i = 0; i < 9; i++) {
        if (count % 2 == 1) {
            for (int k = count - 1; k >= 0; k--) {
                V = V + 1;
                A[x - k][y + k] = V; // 왼쪽 아래로 올라가는 대각선이니
            }
        }
        else {
            for (int k = 0; k < count; k++) {
                V = V + 1;
                A[x - k][y + k] = V; // 오른쪽 위로 올라가는 대각선이니
            }
        }
        if (x == 4) { // 5번째 대각선을 수행하고부터는 x의 값을 건들지 않으니까 계속 해당 if문으로 오게 됨
            y = y + 1;
            count = count - 1;
        }
        else { // 1,2 3,4번째 대각선을 수행하고 여기로 옴
            x = x + 1;
            count = count + 1;
        }
    }

    for (int i = 0; i < 5; i++) { // 배열 출력
        for (int j = 0; j < 5; j++) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }
}
```


(2) - 4. 실행화면

```
Microsoft Visual Studio 디버그 콘솔  
1 3 4 10 11  
2 5 9 12 19  
6 8 13 18 20  
7 14 17 21 24  
15 16 22 23 25  
C:\Users\우은지\source\repos\algorism2\Debug\algorism2.exe(프  
이 창을 닫으려면 아무 키나 누르세요...
```

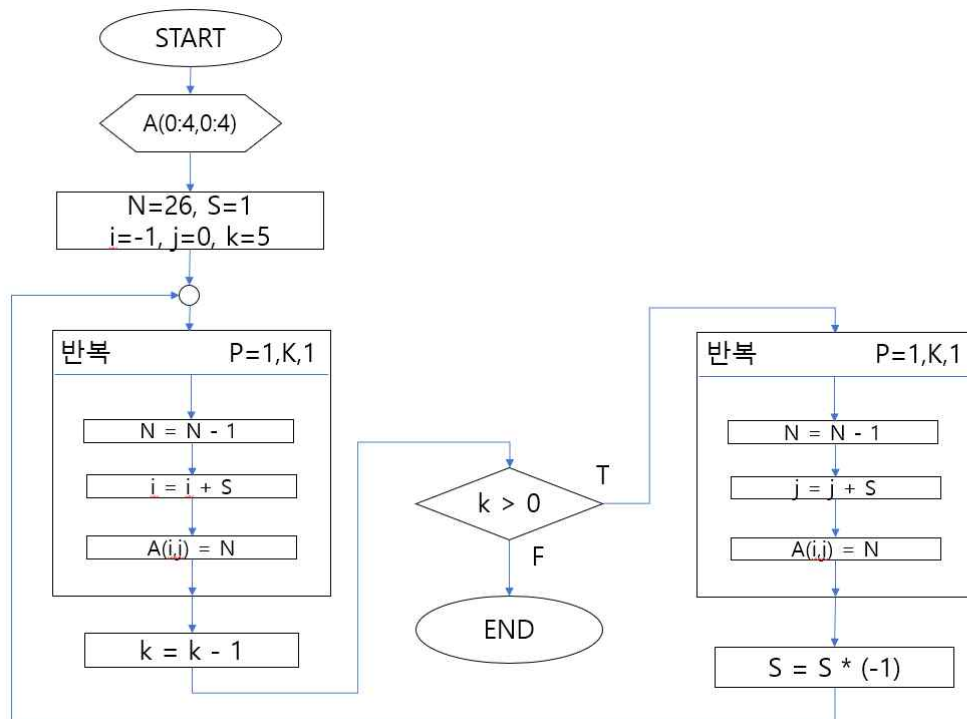
(3) - 1. 문제정의

25	10	11	12	13
24	9	2	3	14
23	8	1	4	15
22	7	6	5	16
21	20	19	18	17

<그림 3>

- 5칸, 4칸, 4칸, 3칸, 3칸, 2칸, 2칸, 1칸씩 같은 방향으로 이동하며 달팽이 모양으로 배열이 채워지도록 해야 한다.
- 이동하는 방향이 행 증가, 열 증가, 행 감소, 열 감소 순으로 반복된다.
- 구현의 편의를 위해 1부터가 아닌 25부터 거꾸로 채운다.

(3) - 2. 설계 (순서도와 설명)



k: 같은 방향으로 한 번에 몇 칸 채울지 조절해 준다.

i: 배열 A의 행을 나타낸다.

j: 배열 A의 열을 나타낸다.

S: 행과 열을 감소 증가를 반복시켜주기 위한 스위치 변수이다.

- 배열의 인덱스를 c언어와 같게 하기 위해서 0번부터 4번 인덱스를 사용하여 배열을 표현한다.
- 행 증가, 열 증가, 행 감소, 열 감소 순으로 반복되도록 하기 위해 S로 증가 감소를 스위치시킨다.
- 왼쪽 for문에서 행 증가, 감소를 담당하고 오른쪽 for문은 열의 증가, 감소를 담당한다.

(3) - 3. 구현 (프로그램 소스)

```
#include <stdio.h>
```

```
void main() {
```

```
    int A[5][5] = { 0 };
```

```
    int N = 26, S = 1, i = -1, j = 0, k = 5;
```

```
    // 25부터 역순으로 넣어주기 위해 N의 초기값을 26으로 설정
```

```
    while (1) {
```

```
        for (int P = 1; P <= k; P++) { // 행 이동
```

```
            N = N - 1;
```

```
            i = i + S;
```

```
            A[i][j] = N;
```

```
        }
```

```
        k = k - 1; // 순서대로 5, 4, 4, 3, 3, 2, 2, 1, 1개씩의 값을 넣어주기 위함
```

```
        if (k <= 0) break;
```

```
        for (int P = 1; P <= k; P++) { // 열 이동
```

```
            N = N - 1;
```

```
            j = j + S;
```

```
            A[i][j] = N;
```

```
        }
```

```
        S = S * (-1); // 달팽이 모양으로 돌기 때문에 행 증가, 열 증가, 행 감소, 열
```

```
        감소가 반복되기 때문에 이를 위해 스위치 변수 S를 사용
```

```
    }
```

```
    for (int i = 0; i < 5; i++) { // 배열 출력
```

```
        for (int j = 0; j < 5; j++) {
```

```
            printf("%d ", A[i][j]);
```

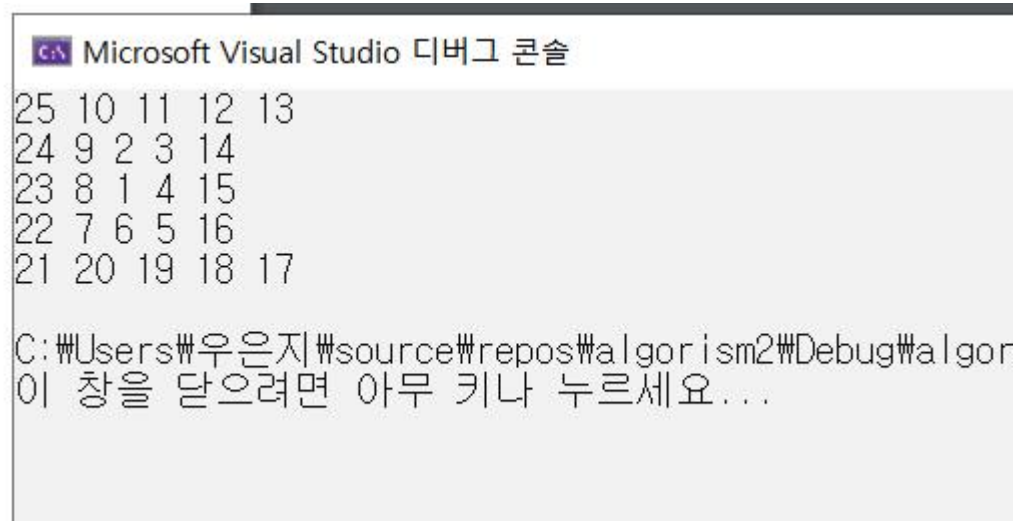
```
        }
```

```
        printf("\n");
```

```
    }
```

```
}
```

(3) - 4. 실행화면



```
Microsoft Visual Studio 디버그 콘솔
25 10 11 12 13
24 9 2 3 14
23 8 1 4 15
22 7 6 5 16
21 20 19 18 17

C:\Users\우은지\source\repos\algorithm2\Debug\algor
이 창을 닫으려면 아무 키나 누르세요...
```

서울여자대학교 카피킬러캠퍼스 표절 검사 결과 확인서

확 인

성 명

서 명

아이디	2020111340	표절률	17%
소속	정보보호학과		
성명	자필로 기재하세요		

검사번호	00148654671	검사일자	2021.11.20 22:52
발급형태	<input checked="" type="checkbox"/> 기본보기 <input type="checkbox"/> 요약보기 <input type="checkbox"/> 상세보기	발급일자	2021.11.20 22:54
검사명	미입력		
문서명	알고리즘_코딩과제2.hwp		
비고			

비교범위	[현재첨부분서] [카피킬러 DB]
검사설정	표절기준 [6 어절], 인용/출처 표시문장 [제외], 법령/경전 포함문장 [제외], 목차/참고문헌 [제외]

검토 의견	
-------	--

분석 정보

표절률	전체문장	동일문장	의심문장	인용/출처	법령/경전
17%	36	0	15	0	0

비교 문서 정보

번호	표절률	출처정보	비고
1	4%	[카피킬러 DB] blog.naver.com - 파일명 : 다차원 배열 - 발행 : naver	
2	4%	[카피킬러 DB] jkpops.blog.me - 파일명 : 빙고 - 발행 : naver	
3	3%	[카피킬러 DB] cafe.naver.com - 파일명 : [10.7.8][허창녕]	
4	3%	[카피킬러 DB] www.nowcoder.com - 파일명 : C翻转__牛客网 - 발행 : www.nowcoder.com	
5	3%	[카피킬러 DB] cosmosscoding.tistory.com - 파일명 : [c 언어] 빙고게임[EASY]	
6	3%	[카피킬러 DB] kin.naver.com - 파일명 : 프린이 C언어 구제 좀 부탁드립니다... ... i++) { printf("%d 번째 사람의 성적을 입력합니다.Wn",i+1); write(i); } printresult(); return 0; } [/code] 일단 실행은 잘 되긴 하지만, arr[4][4]에 arr[4][0]부터 arr[4][3]까지 더한 값을 arr[4][4]에 넣게 되면 원...	
7	2%	[카피킬러 DB] blog.naver.com - 파일명 : 응용 알고리즘(by C,JAVA)_배열(행, 열, 삼각형 모양, 모래시계, 달팽이, 마름...	
8	2%	[카피킬러 DB] blog.naver.com - 파일명 : 210603) 21. 슈팅 게임(오프라인)	
9	2%	[카피킬러 DB] blog.naver.com - 파일명 : [영진전문대학 컴퓨터정보계열]자바 쉬운 문제2 - 발행 : naver	
10	2%	[카피킬러 DB] kin.naver.com - 파일명 : 동적할당후 메모리 해제에서 문제가 생깁니다	
11	1%	[카피킬러 DB] kin.naver.com - 파일명 : c언어 2차원배열관련 질문	
12	1%	[카피킬러 DB] blog.naver.com - 파일명 : 48. 2차 배열 3 (응용,사이클) - 발행 : naver	
13	1%	[카피킬러 DB] yajasu815.blog.me - 파일명 : 중간고사(문자열 입력, 정렬, 검색) 더블 포인터 사용 - 발행 : naver	
14	1%	[카피킬러 DB] cafe.naver.com - 파일명 : [C/C++/썬언어] C언어 소스좀짜주세요!!	
15	1%	[카피킬러 DB] blog.naver.com - 파일명 : 중간고사(문자열 입력, 정렬, 검색) 더블 포인터 사용 - 발행 : naver	

16	1%	[카피킬러 DB] tiny2n.blog.me - 파일명 : for 문에는 왜 '!'를 쓸까 ? - 발행 : naver
17	1%	[카피킬러 DB] kin.naver.com - 파일명 : C언어 소스좀짜주세요!!(내공100완전급함)
18	1%	[카피킬러 DB] yabmoons.tistory.com - 파일명 : [세그먼트 트리(Segment Tree)] 개념과 구현방법 (C++) - 발행 : yabmoons.tistory.com
19	1%	[카피킬러 DB] Copykiller - 파일명 : 도시마케팅 관점의 영상콘텐츠단지 모형개발과 선호유형별 소비자 특성에 관한 연구 - 저자 : 조윤정 - 발행 : 서울 : 홍익대학교 광고홍보대학원, 2006.8
20	1%	[카피킬러 DB] s-space.snu.ac.kr - 파일명 : 과학적 모델의 사회적 구성 수업에서 구현된
21	1%	[카피킬러 DB] www.clien.net - 파일명 : 먹는 것도 중요하지만 싸는 것도 중요하다! : 클리앙
22	1%	[카피킬러 DB] kin.naver.com - 파일명 : 농협 2019년 NCS 기출 문제 해설 질문
23	1%	[카피킬러 DB] blog.naver.com - 파일명 : 빙고 - 발행 : naver
24	1%	[카피킬러 DB] m.blog.naver.com - 파일명 : 1 2 3 4 5 6 7 8 9 10 11 12 우연? 필연? : 네이버 블로그 - 발행 : m.blog.naver.com
25	1%	[카피킬러 DB] kin.naver.com - 파일명 : c언어 5개 숫자입력해서 가로 세로 5개씩
26	1%	[카피킬러 DB] blog.naver.com - 파일명 : C / C 029 : 2차원배열, 지그재그, 달팽이배열 - 발행 : naver
27	1%	[카피킬러 DB] blog.naver.com - 파일명 : C / C 030 : 마방진 - 발행 : naver
28	1%	[카피킬러 DB] blog.naver.com - 파일명 : 마인크래프트 건축 강좌-15.스타크 타워(어벤져스 타워) - 발행 : naver
29	1%	[카피킬러 DB] blog.csdn.net - 파일명 : 迷宫问题算法分析_afeiluo的博客-CSDN博客 - 발행 : blog.csdn.net