

# Level18 카드 종류 알아내기 [난이도 : 3]

문제 1번 [[숙제](#) [목록보기](#)]

cardList 배열에 여러종류의 카드가 있습니다.

아래와 같이 카드를 입력받아주세요 (한 문장을 입력받아주세요, 최대 15글자)

DAT(Direct Address Table)을 이용하여 총 몇 종류의 카드가 있는지 count 해서 출력  
해주세요.

**CardList**

A	B	C	D	A	C	A	B	C	D	E
---	---	---	---	---	---	---	---	---	---	---

출력: 5개

## 입력 예제

ABCDACABCDE

## 출력 결과

5개

```

#include <iostream>
using namespace std;

int main(void)
{
    char CardList[15] = {};
    //for (int i = 0; i < 15; ++i)
    //    cin >> CardList[i];
    cin >> CardList; // for문 사용 안해도 된다.

    int table[128] = {};    // 해쉬 테이블 역할
    bool isEnd = false;
    int len = 0;
    while (!isEnd)
    {
        if (CardList[len] == '\0')
        {
            isEnd = true;
        }
        else
        {
            int idx = CardList[len];
            table[idx] += 1;
            len++;
        }
    }

    int cardSort = 0;
    for (int i = 0; i < 128; ++i)
    {
        if (table[i] != 0)
        {
            cardSort++;
        }
    }

    cout << cardSort << "개";

    return 0;
}

```

# Level18 성실한 직원 찾기 [난이도 : 4]

문제 2번 [[숙제](#) [목록보기](#)]

1~65535 번의 ID를 가진 사람들이 있습니다.

출입기록이 있을 때 가장 성실하게 출근한 사람이 누군지 알려주는 프로그램을 작성해주세요.

(Direct Address Table을 사용해서 풀어주세요, 입력값은 없습니다)

출입기록

65000	35	42	70
70	35	65000	1300
65000	30000	38	42

출력: 65000

## 출력 결과

65000

```

#include <iostream>
using namespace std;

int main(void)
{
    int arr2D[3][4] =
    {
        65000, 35, 42, 70,
        70, 35, 65000, 1300,
        65000, 30000, 38, 42
    };

    int table[65536] = {}; // 1 ~ 65535 의 인덱스값을 가져야한다.
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 4; ++x)
        {
            table[arr2D[y][x]] += 1;
        }
    }
    int maxKey = 1;
    int maxValue = table[1];
    for (int i = 1; i < 65536; ++i)
    {
        if (maxValue < table[i])
        {
            maxKey = i;
            maxValue = table[i];
        }
    }

    cout << maxKey;

    return 0;
}

```

# Level18 안나오는 숫자는? [난이도 : 4]

문제 3번 [[숙제](#) [목록보기](#)]

9개 숫자를 입력받고, 3x3 배열에 채워주세요.

1~9까지 숫자 중 어떤 숫자들이 없는지를 찾아서 출력 해주세요.

ex)

입력:

3 5 5

1 4 9

2 2 1

3	5	5
1	4	9
2	2	1

출력: 6 7 8

(없는 숫자는 6, 7, 8 입니다)

## 입력 예제

3 5 5

1 4 9

2 2 1

## 출력 결과

6 7 8

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

int main(void)
{
    int arr2D[3][3] = {};
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            cin >> arr2D[y][x];
        }
    }

    int table[10] = {};

    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            table[arr2D[y][x]] += 1;
        }
    }

    for (int i = 1; i < 10; ++i)
    {
        if (table[i] == 0)
        {
            cout << i << " ";
        }
    }
    return 0;
}
```

# Level18 인기많은 소년 [난이도 : 4]

문제 4번 [[숙제](#) [목록보기](#)]

8글자를 입력 받아주세요.

가장 많이 나온 알파벳을 출력해주세요.

(1등은 한명만 존재합니다.)

ex)

B	Q	T	A	B	A	B	A
---	---	---	---	---	---	---	---

입력 : BQTABABA

출력: A

## 입력 예제

BQTABAAA

## 출력 결과

A

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

int main(void)
{
    // ASCII 'A' = 65
    // ASCII 'Z' = 90
    char input[8] = {};
    cin >> input;

    int table[91] = {};
    for (int i = 0; i < 8; ++i)
    {
        table[input[i]] += 1;
    }
    int maxIdx = 65; // 'A' = 65
    int maxVal = table[65];

    for (int k = 0; k < 91; ++k)
    {
        if (maxVal < table[k])
        {
            maxVal = table[k];
            maxIdx = k;
        }
    }

    cout << (char)maxIdx;

    return 0;
}
```



## Level18 블랙리스트 [난이도 : 6]

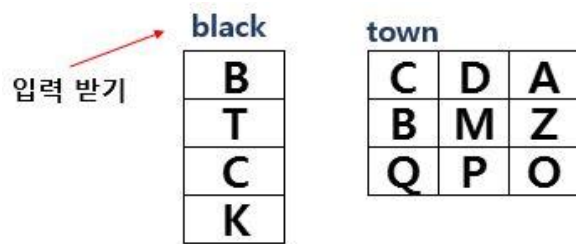
문제 5번 [[숙제](#) [목록보기](#)]

3x3 town배열을 하드코딩 해주세요.

블랙리스트 네명을 입력 받고,

마을에 블랙리스트에 해당하는 사람이 몇 명 있는지 출력하세요.

ex)



입력 : BTCK

출력: 2명

### 입력 예제

BTCK

### 출력 결과

2명

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

int main(void)
{
    char town[3][4] =
    {
        "CDA",
        "BMZ",
        "QPO"
    };

    char black[5] = {};    // 4명
    cin >> black;

    int blackSum = 0;
    for (int i = 0; i < 5; ++i)
    {
        for (int y = 0; y < 3; ++y)
        {
            for (int x = 0; x < 3; ++x)
            {
                if (black[i] == town[y][x])
                {
                    blackSum += 1;
                }
            }
        }
    }

    cout << blackSum << "명";

    return 0;
}
```

# Level18 한줄로 알파벳 정렬하기

[난이도 : 5]

문제 6번 [[숙제](#) [목록보기](#)]

칸에 알파벳들이 채워져 있습니다.

이 알파벳들을 정렬해서 한줄로 출력 해주세요.

(DirectAddressTable를 사용해서 풀어주세요)

A	B	C
A	G	H
H	I	J
K	A	B
A	B	C

출력: AAAABBBCCGHHIJK

## 출력 결과

AAAABBBCCGHHIJK

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

int main(void)
{
    char str2D[5][4] =
    {
        "ABC",
        "AGH",
        "HIJ",
        "KAB",
        "ABC"
    };

    int table[128] = {};    // ASCII : 0 ~ 127
    for (int y = 0; y < 5; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            table[str2D[y][x]] += 1;
        }
    }

    for (int i = 0; i < 128; ++i)
    {
        if (table[i] != 0)
        {
            for (int k = 0; k < table[i]; ++k)
            {
                cout << (char)i;
            }
        }
    }

    return 0;
}
```

# Level18 기차에서 우리팀 찾기

[난이도 : 3]

문제 7번 [[숙제](#) [목록보기](#)]

**train**

3	7	6	4	2	9	1	7
---	---	---	---	---	---	---	---

 ← 하드코딩

**team**

7	6	4
---	---	---

 ← 입력

기차(train)에 탄 사람들 중에 우리팀이 있는지 찾으려 합니다.

우리팀(team)을 입력 받고

우리팀이 기차에서 몇번부터~몇번째 칸에 탑승하고 있는지 출력 하세요.

## 입력 예제

7 6 4

## 출력 결과

1번~3번 칸

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

int main(void)
{
    int train[8] = { 3,7,6,4,2,9,1,7 };
    int input[3] = {};
    cin >> input[0];
    cin >> input[1];
    cin >> input[2];

    int seat[3] = {};
    for (int i = 0; i < 3; ++i)
    {
        for (int k = 0; k < 8; ++k)
        {
            if (input[i] == train[k])
            {
                seat[i] = k;
                break;
            }
        }
    }

    int max = 0;
    int min = 7;
    for (int i = 0; i < 3; ++i)
    {
        if (min > seat[i])
            min = seat[i];
        if (max < seat[i])
            max = seat[i];
    }

    cout << min << "번~" << max << "번 칸";

    return 0;
}
```

## Level18 몇층에 있으세요? [난이도 : 4]

문제 8번 [숙제 [목록보기](#)]

아래는 아파트를 배열으로 나타낸 사진 입니다.

맨 아래는 1층 (15 2 6)

맨 꼭대기층은 5층 (15 18 17) 입니다.

이 배열을 하드코딩 해 주세요.

15	18	17
4	6	9
10	1	3
7	8	9
15	2	6

숫자 3개를 family 배열에 입력 받으세요.

이 family가 몇층에 사는지 isPattern 함수를 이용해서 문제를 풀어 주세요.

### 입력 예제

7 8 9

### 출력 결과

2층

```

#include <iostream>
using namespace std;
void isPattern(int(*apart)[3], int* family)
{
    int layer[5] = {};
    for (int idx = 0; idx < 3; ++idx)
    {
        for (int y = 0; y < 5; ++y)
        {
            for (int x = 0; x < 3; ++x)
            {
                if (apart[y][x] == family[idx])
                {
                    layer[y] += 1;
                }
            }
        }
    }
    int topLayer = 5;
    for (int i = 0; i < 5; ++i)
    {
        if (layer[i] == 3)
            cout << topLayer - i << "층";
    }
}

int main(void)
{
    int apart[5][3] =
    {
        15,18,17,
        4,6,9,
        10,1,3,
        7,8,9,
        15,2,6
    };
    int family[3] = {};
    cin >> family[0];
    cin >> family[1];
    cin >> family[2];

    isPattern(apart, family);

    return 0;
}

```



# Level18 카운팅값 찾기 [난이도 : 4]

문제 9번 [[숙제](#) [목록보기](#)]

1	3	3	5	1
3	6	2	4	2
1	9	2	6	5

위 배열을 하드코딩 하세요.

그리고 숫자 1개를 변수 n에다가 입력 받으세요.

위 배열에는 똑같은 숫자가 여러개가 존재합니다.

예로들어 숫자 1은 3개 존재하고,

숫자6은 2개 존재합니다.

n개 존재하는 숫자를 출력하는 프로그램을 작성 해 주세요

예로들어 숫자 2를 입력받으면,

배열에서 2개 존재하는 숫자를 찾아야 합니다.

따라서 5와 6 을 출력 하시면 됩니다.

## 입력 예제

3

## 출력 결과

1 2 3

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

int main(void)
{
    int arr2D[3][5] =
    {
        1,3,3,5,1,
        3,6,2,4,2,
        1,9,2,6,5
    };
    int n = 0;
    cin >> n;

    int table[10] = {};    // hash table
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 5; ++x)
        {
            table[arr2D[y][x]] += 1;
        }
    }

    for (int i = 0; i < 10; ++i)
    {
        if (table[i] == n)
            cout << i << " ";
    }

    return 0;
}
```