

## Level18.5

Level 16 / 17 때 했던 for문 문제들을 복습하면서 pattern 연습도 해 봅니다.

함수를 쓰면 코딩하기 간편해 지는 문제들인지

DAT를 써서 풀 수 있는 문제인지 파악하는 것도 중요합니다.

앞으로는 for문 문제들이 꾸준히 나오며, **설계 후 코딩**을 하시면 됩니다. 파이팅!

## Level18.5 가족을 찾아라 [난이도 : 3]

문제 1번 [[숙제](#) [목록보기](#)]

G	K	G

**문자 3개**를 배열의 빈 칸에다가 입력 받고, 빈칸에 채워주세요.

총 6개의 문자에서 같은 문자가 3개이상(>=3) 존재 하는지 검사하는 소스코드를 작성해주세요.

존재하면 "있음" 출력

존재하지 않으면 "없음" 출력

예로들어 A B C를 입력받으면

배열에는 G K G A B C 이렇게 6개 문자가 있습니다.

여기서 같은문자 3개이상 존재하는 문자는 없으니

"없음" 이 답입니다.

## 입력 예제

A B C

## 출력 결과

없음

```

#include <iostream>
using namespace std;
// 입력받은 값 중 중복된 값이 있는지 확인해야한다.
// DAT을 사용해서 해결
int main(void)
{
    char str2D[2][4] =
    {
        "GKG",
    };
    cin >> str2D[1][0];
    cin >> str2D[1][1];
    cin >> str2D[1][2];

    int table[128] = {};
    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            table[str2D[y][x]] += 1;
        }
    }
    bool isExist = false;
    for (int i = 0; i < 128; ++i)
    {
        if (table[i] >= 3)
        {
            isExist = true;
        }
    }

    if (isExist)
        cout << "있음";
    else
        cout << "없음";

    return 0;
}

```

# Level18.5 도플갱어 [난이도 : 3]

문제 2번 [[숙제](#) [목록보기](#)]

1	3	3	5	1	4
---	---	---	---	---	---

6개 숫자를 입력 받으세요.

같은 숫자가 존재하는지 확인하고 출력 해주세요.

존재O => "도플갱어 발견" 출력

존재X => "미발견" 출력

**\* 충분한 설계를 하신 후 풀어주시면 됩니다**

## 입력 예제

1 3 3 5 1 4

## 출력 결과

도플갱어 발견

```
#include <iostream>
using namespace std;
// DAT 사용
// index로 사용가능한 경우의수 10개(0 ~ 9)
int main(void)
{
    int arr[6] = {};
    for (int i = 0; i < 6; ++i)
        cin >> arr[i];

    int table[10] = {};    // 0 ~ 9
    for (int k = 0; k < 6; ++k)
        table[arr[k]] += 1;

    bool isExist = false;
    for (int o = 0; o < 10; ++o)
    {
        if (table[o] > 1)
        {
            isExist = true;
            break;
        }
    }

    if (isExist)
        cout << "도플갱어 발견";
    else
        cout << "미발견";

    return 0;
}
```

# Level18.5 알파벳 카운팅 [난이도 : 4]

문제 3번 [[숙제](#) [목록보기](#)]

한문장을 입력 받으세요.(최대 10글자)

그리고 가장 많은 알파벳이 어떤 알파벳인지 출력 해주세요.

## 입력 예제

AKFBBQAAK

## 출력 결과

A

```

#include <iostream>
using namespace std;
// DAT 사용
// index로 사용가능한 경우의수 10개(0 ~ 9)
int main(void)
{
    // A = 65
    // Z = 90
    // A ~ Z = 26개
    char str[11] = {};      // 최대 10글자
    cin >> str;

    int table[27] = {};    // 인덱스 1 ~ 26 을 사용해야함
    for (int i = 0; i < 10; ++i)
    {
        if (str[i] != '\0')
        {
            int idx = str[i] - 64;
            table[idx] += 1;
        }
        else
            break;
    }
    int maxCnt = table[1];
    int maxIdx = 1;
    for (int k = 1; k < 27; ++k)
    {
        if (maxCnt < table[k])
        {
            maxCnt = table[k];
            maxIdx = k;
        }
    }

    cout << (char)(maxIdx + 64);

    return 0;
}

```

### Question)

hash table에는 여러 종류가 있는 것 같던데, 위와 같이 A를 ASCII코드 값 그대로 65 인덱스에 넣지않고 임의로 인덱스값을 코드 작성자 본인의 규칙대로 해서 테이블을 작성하여도 DAT를 사용하여 문제를 해결했다고 할 수 있는가?

## Level18.5 하마의 충치 [난이도 : 2]

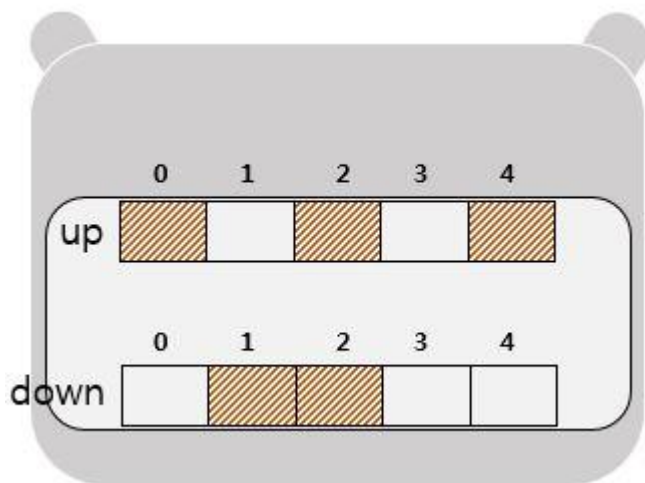
문제 4번 [숙제 목록보기]

하마는 여러충치를 가지고 있는데, 입을 닫으면 이빨끼리 부딪칩니다.

이때 충치끼리 닿으면 하마가 아파합니다.

up 2번치아와 down 2번치아는 충치라서 닿으면 하마가 고통을 느낍니다.

윗니와 아랫니 상태를 입력받고, 고통을 느끼는 치아의 갯수를 출력 하세요.



### 입력 예제

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

### 출력 결과

```
1개
```

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

int main(void)
{
    bool up[5] = {};
    bool down[5] = {};

    for (int i = 0; i < 10; ++i)
    {
        if (i < 5)
            cin >> up[i];
        else
            cin >> down[i - 5];
    }

    int cnt = 0;
    for (int k = 0; k < 5; ++k)
    {
        if (up[k] && down[k])
            cnt += 1;
    }

    cout << cnt << "개";

    return 0;
}
```



# Level18.5 양쪽에서 아이찾기 [난이도 : 3]

문제 5번 [숙제 목록보기]

아래 문장을 하드코딩 해 주세요

**ATKPTCABC**

그리고 두 문자를 입력받아주세요

첫번째 문자는 왼쪽에서부터 오른쪽으로 검색 해 주세요

두번째 문자는 오른쪽에서부터 왼쪽으로 검색 해 주세요

검색해서 가장 먼저 발견되는 글자들이 얼마나 떨어져 있는지 출력하세요

만약 P와 A를 입력받았다면 정답은 3 입니다

**ATKPTCABC**

3칸 떨어짐

## 입력 예제

P A

## 출력 결과

3

```

#include <iostream>
using namespace std;
int main(void)
{
    char str[10] = "ATKPTCABC";
    char input[2] = {};
    cin >> input[0] >> input[1];
    int idx1 = 0, idx2 = 0;
    bool isFind1 = false, isFind2 = false;
    for (int i = 0; i < 2; ++i)
    {
        for(int k = 0; k < 10; ++k)
        {
            if (i == 0)
            {
                if (str[k] == input[i])
                {
                    idx1 = k;
                    isFind1 = true;
                    break;
                }
            }
            else
            {
                int k2 = 9 - k;
                if (str[k2] == input[i])
                {
                    idx2 = k2;
                    isFind2 = true;
                    break;
                }
            }
        }
    }
    if (isFind1 && isFind2)
    {
        int len = idx2 - idx1;
        cout << len;
    }
    else
        cout << "입력한 문자중 존재하지 않는 문자가 있습니다.";

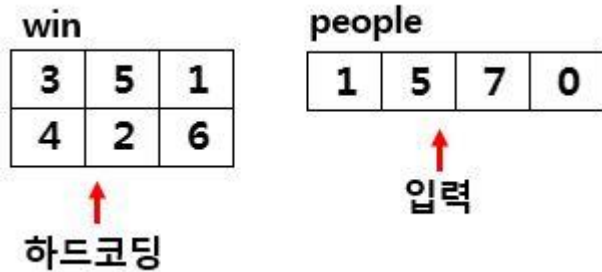
    return 0;
}

```

# Level18.5 사법고시 합격자 발표

[난이도 : 3]

문제 6번 [[숙제](#) [목록보기](#)]



win 배열에 합격자 명단이 있습니다.

people의 번호를 입력 받고,

네명의 합격 여부를 출력 해주세요.

## 입력 예제

1 5 7 0

## 출력 결과

1번 합격

5번 합격

7번 불합격

0번 불합격

```

#include <iostream>
using namespace std;

int main(void)
{
    int win[2][3] =
    {
        3,5,1,
        4,2,6
    };
    int people[4] = {};
    for (int i = 0; i < 4; ++i)
        cin >> people[i];

    int table[10] = {};    // 0번~9번
    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            table[win[y][x]] += 1;
        }
    }
    bool isWin[4] = {};
    for (int k = 0; k < 4; ++k)
    {
        if(table[people[k]] == 1)
            isWin[k] = true;
    }

    for (int o = 0; o < 4; ++o)
    {
        cout << people[o] << "번 ";
        if (isWin[o])
            cout << "합격\n";
        else
            cout << "불합격\n";
    }

    return 0;
}

```

# Level18.5 민코딩 찾기 [난이도 : 5]

문제 7번 [[숙제](#) [목록보기](#)]

"**MINCODING**" 문장을 **vect**라는 배열에 하드코딩 해 주세요

이제 찾을 문자 개수 **n**과 문자들을 입력 받습니다.

입력받은 각 문자가 **vect**배열에 존재하는지 출력 해 주세요.

존재하면 0, 존재하지 않으면 X를 출력 해 주세요.

**\* DAT(Direct Address Table)을 이용하시면 됩니다**

ex)

5

A B C D E

출력결과 : XX00X

## 입력 예제

3

I N G

## 출력 결과

000

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

int main(void)
{
    char vect[10] = "MINCODING";
    int len = 0;
    char input[10] = {};
    cin >> len;
    for (int i = 0; i < len; ++i)
        cin >> input[i];

    int table[128] = {};    // hash table 생성
    for (int k = 0; k < 10; ++k)
        table[vect[k]] += 1;

    for (int o = 0; o < len; ++o)
    {
        if (table[input[o]] > 0)
            cout << "0";
        else
            cout << "X";
    }

    return 0;
}
```

# Level18.5 다른 문장 입력 하기

## [난이도 : 5]

문제 8번 [[숙제](#) [목록보기](#)]

세 문장을 입력받고

같은글자가 한글자라도 없으면 Perfect

아니면 No를 출력 해 주세요.

Ex)

ABCDEF

GHIJKLMN

OPQR

를 입력 하였으면

**Perfect**를 출력 해 주세요.

Ex2)

BBQWORLD

ABCWORLD

ZYM

를 입력 하였으면 **No**를 출력 해 주세요.

## 입력 예제

ABCDEF

GHIJKLMN

OPQR

## 출력 결과

Perfect

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

int main(void)
{
    char str2D[3][20] = {};
    cin >> str2D[0];
    cin >> str2D[1];
    cin >> str2D[2];

    int table[128] = {};
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 20; ++x)
        {
            if (str2D[y][x] == '\\0')
                break;
            else
                table[str2D[y][x]] += 1;
        }
    }

    bool isDouble = false;
    for (int i = 65; i < 91; ++i)
    {
        if (table[i] > 1)
        {
            isDouble = true;
            break;
        }
    }

    if (isDouble)
        cout << "No";
    else
        cout << "Perfect";

    return 0;
}
```



# Level18.5 중복 제거하기 [난이도 : 4]

문제 9번 [[숙제 목록보기](#)]

한문장을 입력 받고, 중복 알파벳을 제거한 후 알파벳 순서대로 출력 해주세요.

(A~F글자, 최대 10글자, **direct address table** 자료구조를 이용해주세요)

ex)

입력 : ABBACCDEA

출력 : ABCDE

## 입력 예제

ABBACCDEA

## 출력 결과

ABCDE

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

int main(void)
{
    char str[11] = {};          // 최대 10글자
    cin >> str;
    // ascii -> A : 65, Z = 90
    int table[26] = {};        // idx = 알파벳 - 65
    for (int i = 0; i < 11; ++i)
    {
        if (str[i] == '\0')
            break;
        else
        {
            table[str[i] - 65] += 1;
        }
    }

    for (int k = 0; k < 26; ++k)
    {
        if (table[k] > 0)
            cout << (char)(k + 65);
    }

    return 0;
}
```

# Level18.5 각 글자수 세기 [난이도 : 3]

문제 10번 [[숙제](#) [목록보기](#)]

한문장을 입력받고, 각 글자마다 수를 출력 해주세요.

최대 10글자 까지 입력될 수 있습니다

(DirectAddressTable을 이용해주세요)

ex)

입력 : BTABATP

출력:

A:2

B:2

P:1

T:2

## 입력 예제

BTABATP

## 출력 결과

A:2

B:2

P:1

T:2

```

#include <iostream>
using namespace std;

int main(void)
{
    char str[11] = {};          // 최대 10글자
    cin >> str;

    // ascii -> A = 65, Z = 90
    int table[26] = {};        // index = 알파벳 - 65
    for (int i = 0; i < 10; ++i)
    {
        if (str[i] == '\0')
            break;
        else
            table[str[i] - 65] += 1;
    }

    for (int i = 0; i < 26; ++i)
    {
        if (table[i] > 0)
        {
            cout << (char)(i + 65) << ":";
            cout << table[i] << endl;
        }
    }
    return 0;
}

```

# Level18.5 유령은 존재 할까? [난이도 : 3]

문제 11번 [[숙제 목록보기](#)]

한 문장을 입력 받으세요.

한 문장에서 GHOST 단어가 존재하는지 찾아서 출력 해 주세요.

QGHOSTA를 입력 받았다면

입력받은 문장안에 GHOST 가 존재 하므로 **존재**를 출력 해 주세요.

ABGOSAT를 입력 받았다면

입력받은 문장안에 GHOST가 존재 하지 않으므로 **존재하지 않음** 을 출력 해 주세요.

## 입력 예제

QGHOSTA

## 출력 결과

존재

```

#include <iostream>
using namespace std;

void check(char* source, int idx, bool* isExist)
{
    char str[6] = "GHOST";
    int chkCnt = 0;
    for (int i = 0; i < 5; ++i)
    {
        if (source[idx] == str[i])
        {
            chkCnt += 1;
            idx++;
        }
    }
    if (chkCnt == 5)
        *isExist = true;
}

int main(void)
{
    char str[20] = {};
    cin >> str;

    int gList[20] = {};
    int idx = 0;
    for (int i = 0; i < 20; ++i)
    {
        if (str[i] == '\\0')
            break;
        else
        {
            if (str[i] == 'G')
            {
                gList[idx] = i;
                idx += 1;
            }
        }
    }
    bool isExist = false;
    for (int k = 0; k < idx; ++k)
    {
        int idx = gList[k];
        if (str[idx] == '\\0')

```

```
                break;
            else
            {
                check(str, idx, &isExist);
                if (isExist)
                    break;
            }
        }

        if (isExist)
            cout << "존재";
        else
            cout << "존재하지 않음";

        return 0;
    }
}
```