

Level17 offset으로 범인을 찾아라

[난이도 : 2]

문제 1번 [[숙제](#) [목록보기](#)]

아래의 2차배열을 하드코딩 해 주세요.

A	T	K	B
C	Z	F	D
H	G	E	I

문자 1개를 입력받고

y, x축의 offset 값을 입력받습니다.

입력받은 문자 위치에서 offset만큼 떨어져 있는 곳의 문자를 출력 하세요.

만약 T 2 1를 입력받았다면,

T에서 y축으로 + 2 , x축으로 + 1만큼 떨어진

알파벳인 E를 출력하시면 됩니다.

입력 예제

T 2 1

출력 결과

E

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

int main(void)
{
    char str2D[3][5] =
    {
        "ATKB",
        "CZFD",
        "HGEI"
    };
    char input = '\0';
    cin >> input;
    int offsetX = 0, offsetY = 0;
    cin >> offsetY >> offsetX;
    char target = '\0';

    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 4; ++x)
        {
            if (str2D[y][x] == input)
            {
                target = str2D[y + offsetY][x + offsetX];
                break;
            }
        }
        if (target != '\0')
        {
            break;
        }
    }

    cout << target;

    return 0;
}
```

Level17 비트배열 마스크처리와 합 구하기

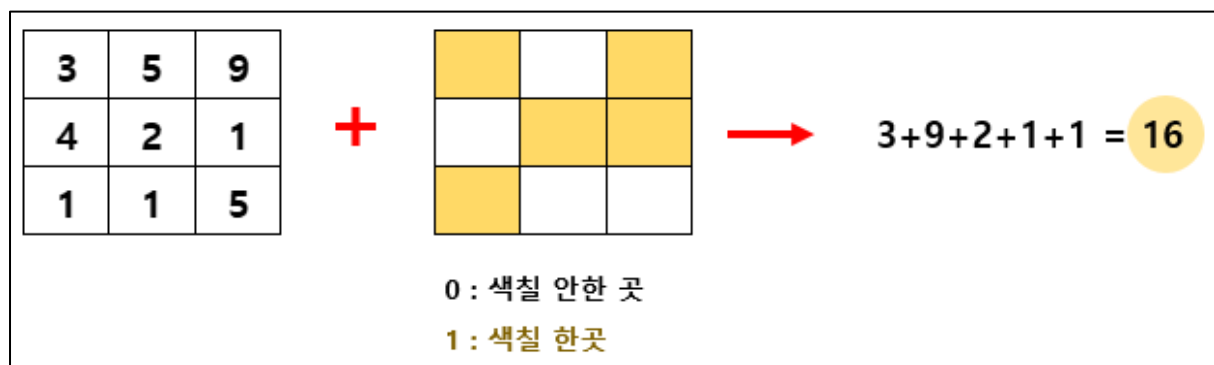
[난이도 : 3]

문제 2번 [[숙제](#) [목록보기](#)]

숫자배열을 하드코딩 해주세요.

그리고 matching 되는 3x3 비트배열을 입력 받으세요.

masking 처리 후 나온 숫자들의 합을 출력하세요.



입력 예제

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

출력 결과

```
16
```

```

#include <iostream>
using namespace std;

int main(void)
{
    int arr2D[3][3] =
    {
        3,5,9,
        4,2,1,
        1,1,5
    };

    int arrBit[3][3] = {};
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            cin >> arrBit[y][x];
        }
    }
    int sum = 0;
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            if (arrBit[y][x] == 1) {
                sum += arr2D[y][x];
            }
        }
    }

    cout << sum;

    return 0;
}

```

Question)

비트값 배열을 입력받는 방법이 있는가?

Level17 네가 있는 거리에서 [난이도 : 3]

문제 3번 [숙제 목록보기]

아래 배열을 하드코딩 하고, 숫자 2개를 입력받습니다.

5	9	4	6	1	5	8	9
---	---	---	---	---	---	---	---

첫번째 숫자는 기준점 index 이고,

두번째 숫자는 찾을 Target 숫자 값입니다.

기준점 index로 부터 오른쪽으로 Target 숫자를 찾습니다.

가장 가까운 거리에 있는 Target을 찾아, 그 Offset값을 출력 해 주세요

* Offset : 기준점에서 얼마나 떨어져 있는지를 나타내는 값

예로들어 3과 8을 입력받았다면

3번 index에서부터 오른쪽으로 숫자 8이 어디있는지 찾아야합니다.

가장 가까운 8은 세 번째 떨어진 곳에 존재하므로,

Offset값은 3 입니다.

입력 예제

1 6

출력 결과

2

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

int main(void)
{
    int arr[8] = { 5,9,4,6,1,5,8,9 };
    int idx = 0, target = 0;
    cin >> idx >> target;
    int offset = 0;
    bool isFind = false;
    for (int i = idx; i < 8; ++i)
    {
        if (arr[i] == target)
        {
            offset = i - idx;
            isFind = true;
            break;
        }
    }
    if (isFind)
    {
        cout << offset;
    }
    else
    {
        cout << "없다.";
    }

    return 0;
}
```

Level17 isExist함수로 보물찾기

[난이도 : 3]

문제 4번 [[숙제](#) [목록보기](#)]

아래 배열을 하드코딩 해 주세요

그리고 숫자 3개를 **배열에다가 입력받아주세요**

3	5	9
4	2	1
5	1	5

isExist함수를 이용해서 각각 입력받은 숫자가

하드코딩한 배열안에 존재하는지 출력 해 주세요

입력 예제

3 7 1

출력 결과

3:존재

7:미발견

1:존재

```

#include <iostream>
using namespace std;

void isExist(int (*arr)[3], int* input) // argument로 int arr[3][3]도 가능하다.
{
    int arrExit[3] = {};
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            if (arr[y][x] == input[x])
            {
                arrExit[x]++;
            }
        }
    }

    for (int i = 0; i < 3; ++i)
    {
        cout << input[i] << ":";
        if (arrExit[i] > 0)
        {
            cout << "존재\n";
        }
        else
        {
            cout << "미발견\n";
        }
    }
}

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

    isExist(arr2D, arrInput);

    return 0;
}

```

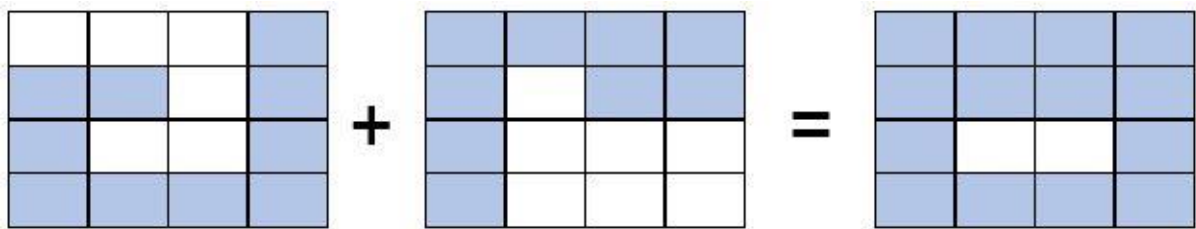

Level17 합친 mask배열의 좌표 구하기

[난이도 : 4]

문제 5번 [[숙제](#) [목록보기](#)]

아래 두 mask 배열을 합쳐주세요.

그리고 비어있는 곳의 좌표를 출력하세요.



출력 결과

(2,1)

(2,2)

```

#include <iostream>
using namespace std;

int main(void)
{
    int arr2D_1[4][4] =
    {
        0,0,0,1,
        1,1,0,1,
        1,0,0,1,
        1,1,1,1
    };

    int arr2D_2[4][4] =
    {
        1,1,1,1,
        1,0,1,1,
        1,0,0,0,
        1,0,0,0
    };

    int arrSum[4][4] = {};

    for (int y = 0; y < 4; ++y)
    {
        for (int x = 0; x < 4; ++x)
        {
            arrSum[y][x] = arr2D_1[y][x] + arr2D_2[y][x];
        }
    }

    for (int y = 0; y < 4; ++y)
    {
        for (int x = 0; x < 4; ++x)
        {
            if (arrSum[y][x] == 0)
            {
                cout << "(" << y << "," << x << ")\n";
            }
        }
    }

    return 0;
}

```

Level17 마스킹 처리후의 입력값 존재 여부

[난이도 : 4]

문제 6번 [[숙제](#) [목록보기](#)]

3	5	4	1	1
3	5	2	5	6

> 둘 다 하드코딩

위 두 배열을 하드코딩 해주세요.

숫자 1개를 입력하고

masking 처리된 곳 중 입력 받은 숫자가 존재하는지 안하는지를 출력 해주세요.

ex)

- 5를 입력 하면 존재하기 때문에 "5 존재"
- 1을 입력 하면 없기 때문에 "1 없음"

입력 예제

1

출력 결과

1 없음

```

#include <iostream>
using namespace std;

int main(void)
{
    // true = 1, false = 0
    bool arr_1[2][5] =
    {
        0,0,1,0,0,
        0,0,1,1,1
    };

    int arr_2[2][5] =
    {
        3,5,4,1,1,
        3,5,2,5,6
    };

    int input = 0;
    cin >> input;
    bool isExist = false;

    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 5; ++x)
        {
            if (arr_2[y][x] == input)
            {
                if (arr_1[y][x])
                {
                    isExist = true;
                }
            }
        }
    }

    if (isExist)
        cout << input << " 존재";
    else
        cout << input << " 없음";

    return 0;
}

```

Level17 마을사람들 찾기 [난이도 : 5]

문제 7번 [숙제 목록보기]

아래와 같이 1차 배열을 하드코딩 해 주세요.

그리고 문자 4개를 target 이라는 배열에 입력 받으세요

vect

B	T	K	I	G	Z
---	---	---	---	---	---

target

--	--	--	--

입력받은 문자 중 총 몇 개의 알파벳이 vect배열 안에 존재하는지 출력 하세요.

예를들어,

A B Y Z를 입력 하면, B와 Z가 존재하기 때문에 답은 2 입니다.

예를들어,

T K G F를 입력하시면, 숫자 3이 출력 됩니다.

입력 예제

A B Y Z

출력 결과

2

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

int main(void)
{
    char vect[7] = "BTKIGZ";
    char target[4] = {};
    for (int i = 0; i < 4; ++i)
        cin >> target[i];

    int sum = 0;

    for (int i = 0; i < 4; ++i)
    {
        for (int k = 0; k < 7; ++k)
        {
            if (target[i] == vect[k])
            {
                sum++;
                break;
            }
        }
    }

    cout << sum;
    return 0;
}
```

Level17 인기투표 [난이도 : 4]

문제 8번 [[숙제](#) [목록보기](#)]

아래의 2차배열을 하드코딩 해 주세요.

vect

3	7	4
2	2	4
2	2	5

target

--	--	--

숫자 3개를 target 배열에 입력받아 주세요.

각각 vect에 몇개가 있는지 counting 하고, 이중 가장 빈도수가 높은 숫자를 출력 하세요.

ex)

2 3 4

2는 4개, 3은 1개, 4는 2개가 있으므로

2 를 출력 하시면 됩니다.

입력 예제

2 3 4

출력 결과

2

```

#include <iostream>
using namespace std;

int main(void)
{
    int vect[3][3] =
    {
        3,7,4,
        2,2,4,
        2,2,5
    };
    int target[3] = {};
    cin >> target[0];
    cin >> target[1];
    cin >> target[2];

    int count[3] = {};
    for (int i = 0; i < 3; ++i)
    {
        for (int y = 0; y < 3; ++y)
        {
            for (int x = 0; x < 3; ++x)
            {
                if (target[i] == vect[y][x])
                    count[i]++;
            }
        }
    }

    int maxCount = count[0];
    int maxIdx = 0;
    if (maxCount < count[1])
    {
        maxCount = count[1];
        maxIdx = 1;
    }
    if (maxCount < count[2])
    {
        maxCount = count[2];
        maxIdx = 2;
    }

    cout << target[maxIdx];

    return 0;
}

```


Level17 합격자 발표일 [난이도 : 4]

문제 9번 [[숙제](#) [목록보기](#)]

합격자 발표가 났습니다.

합격자 명단인 아래의 배열을 하드코딩 해 주세요.

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

이제, 존재여부를 찾을 숫자 4개를 univer 배열에 입력 받으세요.

univer

입력받은 숫자들이 존재하면 OK, 없으면 NO를 예제와 같이 출력 해 주세요.

(isExist 함수를 써서 풀어 주세요)

입력 예제

1 3

5 9

출력 결과

OK OK

NO NO

```

#include <iostream>
using namespace std;

void isExist(int* arr, int(*univer)[2])
{
    bool check[2][2] = {};
    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 2; ++x)
        {
            for (int idx = 0; idx < 6; ++idx)
            {
                if (univer[y][x] == arr[idx])
                {
                    check[y][x] = 1;
                    break;
                }
            }
        }
    }
    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 2; ++x)
        {
            if (check[y][x])
                cout << "OK ";
            else
                cout << "NO ";
        }
        cout << endl;
    }
}

int main(void)
{
    int arr[6] = { 3,7,4,1,2,6 };
    int univer[2][2] = {};
    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 2; ++x)
            cin >> univer[y][x];
    }

    isExist(arr, univer);

    return 0;
}

```

Level17 저격 SUM [난이도 : 3]

문제 10번 [[숙제](#) [목록보기](#)]

아래의 배열을 하드코딩 해 주세요.

3	4	1	1	2	6	8	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

getSum(int index)



getSum 함수를 위와 같이 정의 해 주세요.

startIndex를 입력 받고 getSum 함수를 이용해서

숫자 다섯개의 합을 출력 해 주세요.

입력 예제

1

출력 결과

14

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

int arr[11] = { 3,4,1,1,2,6,8,7,8,9,10 };

int getSum(int index)
{
    int sum = 0;
    for (int i = 0; i < 5; ++i)
    {
        sum += arr[index + i];
    }

    return sum;
}

int main(void)
{
    int input = 0;
    cin >> input;

    cout << getSum(input);

    return 0;
}
```

Level17 동명이인 [난이도 : 3]

문제 11번 [[숙제](#) [목록보기](#)]

두명의 이름을 입력 받으세요.

두명의 이름이 완전히 같으면 "동명" 다르면 "남남" 으로 출력 해주세요.

(isSame 함수를 이용)



입력 예제

jason

jason

출력 결과

동명

```

#include <iostream>
using namespace std;

void isSame(char* str1, char* str2)
{
    int len1 = 0;
    int len2 = 0;
    bool isEnd = false; // 0
    while (!isEnd)
    {
        if (str1[len1] == '\0')
            isEnd = true;
        else
            len1++;
    }
    isEnd = false;
    while (!isEnd)
    {
        if (str2[len2] == '\0')
            isEnd = true;
        else
            len2++;
    }

    bool isSame = true;
    if (len1 == len2)
    {
        for (int i = 0; i < len1; ++i)
        {
            if (str1[i] != str2[i])
            {
                isSame = false;
                break;
            }
        }
    }
    else
    {
        isSame = false;
    }

    if (isSame)
        cout << "동명";
    else
        cout << "남남";
}

```

```
int main(void)
{
    char name1[20] = {};
    char name2[20] = {};

    cin >> name1;
    cin >> name2;

    isSame(name1, name2);

    return 0;
}
```