

Level17.5

복잡한 for문을 이용한 문제를 풀 때는, 어떻게 코딩을 할 지 세부 설계가 필요합니다.

코딩을 시작하기 전, 아래 예시와 같은 내용들을 종이에 적으면서 계획을 세워주세요.

1. 몇 중 for문을 써야 할지,
2. for문을 정확히 몇 부터 시작해서 몇 까지 돌릴지,
3. 어떤 조건문이 필요할지?
4. 설계한 대로 for문이 잘 동작 될 지 시뮬레이션 해보기

이렇게 세부적인 설계가 완료만 된다면, 코딩 난이도가 확실히 줄어들게 됩니다.

for문을 보다 편안하게 쓸 수 있도록 세부설계를 하면서 문제를 풀어주세요.

Level17.5 입력받은 비트배열로 마스크하기

[난이도 : 4]

문제 1번 [[숙제](#) [목록보기](#)]

아래에 그림처럼 vect배열을 하드코딩 하고, 비트배열을 입력 받으세요.

마스킹 한 후 나오는 값을 7로 바꾸어 출력 해주세요

bit	0	1	1	0	1	1	0
-----	---	---	---	---	---	---	---

vect	3	5	4	2	6	6	5
------	---	---	---	---	---	---	---

↓ 마스크

0	5	4	0	6	6	0
---	---	---	---	---	---	---

↓

0	7	7	0	7	7	0
---	---	---	---	---	---	---

출력결과 : 0770770

입력 예제

0 1 1 0 1 1 0

출력 결과

0770770

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

int main(void)
{
    bool bit[7] = {};
    int vect[7] = {3,5,4,2,6,6,5};

    for (int i = 0; i < 7; ++i)
    {
        cin >> bit[i];
    }

    for (int k = 0; k < 7; ++k)
    {
        if (bit[k])
            vect[k] = 7;
        else
            vect[k] = 0;
    }

    for (int o = 0; o < 7; ++o)
        cout << vect[o];

    return 0;
}
```

Level17.5 금고털이 [난이도 : 3]

문제 2번 [[숙제](#) [목록보기](#)]



password

3	7	4	9
---	---	---	---

금고 비밀번호는 3 7 4 9 입니다.

그리고 네 자리 비밀번호를 input 배열에 입력받고,

완전히 동일한 비밀번호인지 `isSame` 함수를 이용해서 풀어주세요.

동일하면 "pass", 그렇지 않으면 "fail" 로 출력 해주세요.

입력 예제

3 7 4 9

출력 결과

pass

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

void isSame(int* pw, int* input)
{
    bool isSame = true;
    for (int i = 0; i < 4; ++i)
    {
        if (pw[i] != input[i])
        {
            isSame = false;
            break;
        }
    }
    if (isSame)
        cout << "pass";
    else
        cout << "fail";
}

int main(void)
{
    int password[4] = { 3,7,4,9 };
    int input[4] = {};

    for (int i = 0; i < 4; ++i)
        cin >> input[i];

    isSame(password, input);

    return 0;
}
```

Level17.5 도와주세요 다이어트 [난이도: 5]

문제 3번 [숙제 목록보기]

다이어트를 시작하기 위해 칼로리별로 과일이 몇 개 있는지 counting하려고 합니다.

먼저 levelTable 배열을 하드코딩 해 주세요

levelTable		
단계0	10	20
단계1	30	60
단계2	100	150
단계3	200	300

levelTable에서

lev0(단계0)은 $10 \leq x \leq 20$ 사이의 칼로리를,

lev1(단계1)은 $30 \leq x \leq 60$ 사이의 칼로리를,

lev2(단계2)는 $100 \leq x \leq 150$ 사이의 칼로리를,

lev3(단계3)은 $200 \leq x \leq 300$ 사이의 칼로리를 뜻합니다.

이제 과일 6개의 칼로리를 입력 받으세요. (숫자 6개 입력)

단계0부터 단계3까지 각 단계마다

과일이 몇 개 있는지 Counting 후 출력 해 주세요.

입력 예제

15 15 15 40 100 105

출력 결과

lev0:3

lev1:1

lev2:2

lev3:0

```

#include <iostream>
using namespace std;

int main(void)
{
    int levelTable[4][2] =
    {
        10, 20,
        30, 60,
        100, 150,
        200, 300
    };

    int inputCal[6] = {};
    for (int i = 0; i < 6; ++i)
        cin >> inputCal[i];

    int levelCount[4] = {};
    for (int y = 0; y < 4; ++y)
    {
        for (int idx = 0; idx < 6; ++idx)
        {
            if (levelTable[y][0] <= inputCal[idx] && inputCal[idx] <=
levelTable[y][1])
                levelCount[y]++;
        }
    }

    for (int i = 0; i < 4; ++i)
    {
        cout << "lev" << i << ":" << levelCount[i] << endl;
    }

    return 0;
}

```

Level17.5 컬러찾기 [난이도 : 4]

문제 4번 [숙제 목록보기]

map 배열을 하드코딩하고,

pix 배열에 숫자 4개를 입력받으세요

pix배열에 있는 색상값들이 map에 존재하는 컬러인지 확인해주는 프로그램을 작성해주세요.

만약 pix배열에

1 10

55 -5

값이 입력되었다면, 아래 그림같이 배열에 값이 채워집니다.

map			pix	
3	55	42	1	10
-5	-9	-10	55	-5

pix 배열에 있는 컬러가 map에 있다면 Y를, 없다면 N을 출력하면 됩니다.

ex)

pix				
1	10	→	N	N
55	-5		Y	Y
pix				
42	43	→	Y	N
-7	-9		N	Y

입력 예제

1 10

55 -5

출력 결과

N	N
Y	Y

```

#include <iostream>
using namespace std;

int main(void)
{
    int map[2][3] =
    {
        3, 55, 42,
        -5, -9, -10
    };
    int pix[2][2] = {};
    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 2; ++x)
            cin >> pix[y][x];
    }

    int arr[4] = {};
    int idx = 0;
    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 2; ++x)
        {
            arr[idx] = pix[y][x];
            idx++;
        }
    }

    bool check[4] = {};
    for (int i = 0; i < 4; ++i)
    {
        bool isFind = false;
        for (int y = 0; y < 2; ++y)
        {
            for (int x = 0; x < 3; ++x)
            {
                if (arr[i] == map[y][x])
                {
                    check[i] = 1;
                    isFind = true;
                    break;
                }
            }
        }
    }
}

```

```
        if (isFind)
        {
            break;
        }
    }
}

for (int k = 0; k < 4; ++k)
{
    if (check[k])
        cout << "Y ";
    else
        cout << "N ";

    if (k == 1)
        cout << endl;
}

return 0;
}
```

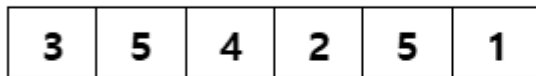
Level17.5 비트배열에서 min값의 인덱스와 값 구하기 [난이도 : 4]

문제 5번 [[숙제](#) [목록보기](#)]

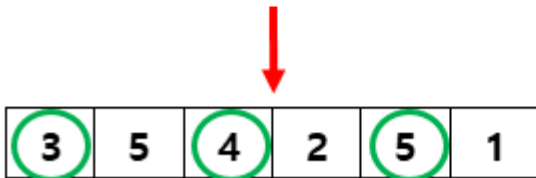
arr배열에 숫자 6개를 입력 받습니다.

masking 처리 후 남은값들 중에서

min값을 찾고 min값의 index를 출력 하세요.



← 숫자 6개 입력



입력 예제

3 5 4 2 5 1

출력 결과

arr[0]=3

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

int main(void)
{
    int arr[6] = {};
    for (int i = 0; i < 6; ++i)
        cin >> arr[i];

    int min = arr[0];
    int minIdx = 0;
    for (int i = 0; i < 6; ++i)
    {
        if (i % 2 == 0)
        {
            if (min > arr[i])
            {
                min = arr[i];
                minIdx = i;
            }
        }
    }

    cout << "arr[" << minIdx << "]= " << min;

    return 0;
}
```

Level17.5 masking 후 범위안의 숫자 존재여부 구하기 [난이도 : 3]

문제 6번 [[숙제](#) [목록보기](#)]

masking 처리 후, 3부터 5사이의 숫자가 존재하는지 출력.

(3 <= x <= 5)

↓ 하드코딩하기

3	1	9
7	2	1
1	0	8

↓ 입력받기

ex)

존재하면 "발견"

존재하지 않으면 "미발견"

입력 예제

1 1 1

1 0 0

1 0 0

출력 결과

발견

```

#include <iostream>
using namespace std;

int main(void)
{
    int arr2D[3][3] =
    {
        3,1,9,
        7,2,1,
        1,0,8
    };

    bool arrInput[3][3] = {};
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            cin >> arrInput[y][x];
        }
    }
    bool isFind = false;
    for (int y = 0; y < 3; ++y)
    {
        for (int x = 0; x < 3; ++x)
        {
            if (arrInput[y][x])
            {
                if (arr2D[y][x] >= 3 && arr2D[y][x] <= 5)
                {
                    isFind = true;
                    break;
                }
            }
        }
        if (isFind)
            break;
    }
    if (isFind)
        cout << "발견";
    else
        cout << "미발견";

    return 0;
}

```

Level17.5 비트배열로 마스킹 처리하기

[난이도 : 3]

문제 7번 [숙제 목록보기]

3	5	4	2
---	---	---	---

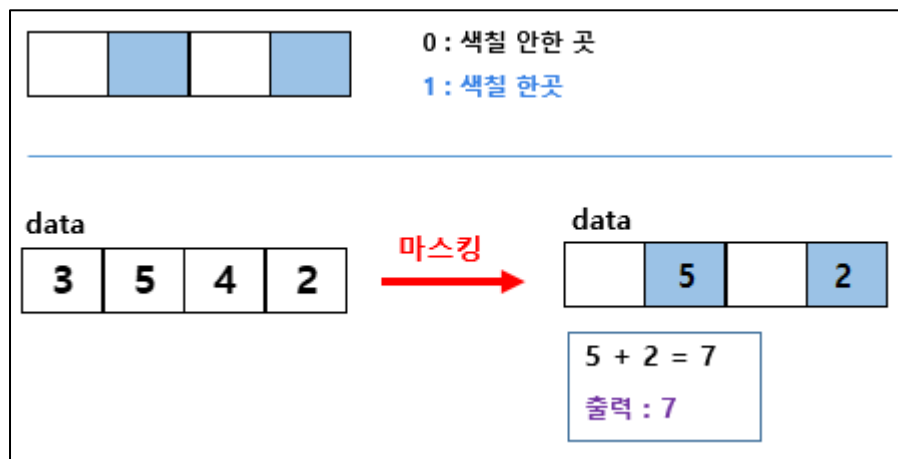
위 배열을 하드코딩하고,

네 자리 비트배열을 입력 받으세요.

마스킹 처리 후, 남은 숫자들의 합을 구해서 출력하면 됩니다.

예를들어

0 1 0 1 을 입력했다면, 다음과 같이 그릴 수 있습니다.



입력 예제

0 1 0 1

출력 결과

7


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

int main(void)
{
    int arr[4] = { 3,5,4,2 };
    bool arrBit[4] = {};
    for (int i = 0; i < 4; ++i)
        cin >> arrBit[i];

    int sum = 0;
    for (int i = 0; i < 4; ++i)
    {
        if (arrBit[i])
        {
            sum += arr[i];
        }
    }

    cout << sum;

    return 0;
}
```

Level17.5 레이더 만들기 [난이도 : 3]

문제 8번 [[숙제](#) [목록보기](#)]

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

문자 1개를 입력 받으세요.

해당 알파벳이 'M'(2,2)로 부터 얼마나 떨어져있는지 offset값을 출력 해주세요.

(offset : y,x 기준)

예를들어

N은 0,1 만큼 떨어져있고,

H는 -1,0 만큼 떨어져있습니다.

입력 예제

F

출력 결과

-1,-2

```

#include <iostream>
using namespace std;

int main(void)
{
    char arr2D[5][5] = {};

    char val = 'A';
    for (int y = 0; y < 5; ++y)
    {
        for (int x = 0; x < 5; ++x)
        {
            arr2D[y][x] = val;
            val++;
        }
    }

    char input = '\0';
    cin >> input;

    int inputX = 0;
    int inputY = 0;
    bool isFind = false;
    for (int y = 0; y < 5; ++y)
    {
        for (int x = 0; x < 5; ++x)
        {
            if (arr2D[y][x] == input)
            {
                inputX = x;
                inputY = y;
                isFind = true;
                break;
            }
        }
        if (isFind)
            break;
    }

    int offsetY = inputY - 2;
    int offsetX = inputX - 2;
    cout << offsetY << ", " << offsetX;

    return 0;
}

```