

환영합니다. 임베디드 전문가 그룹 월택에서 운영하는 비공개 온라인 채점서버입니다.

문제 A2: [SOL] 7 segment

실행시간 제한: 1 Sec 메모리사용 제한: 128 MB

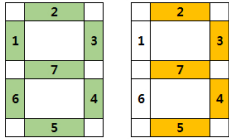
제출: 3 통과: 66.7%

[\[제출\]](#)

문제 설명

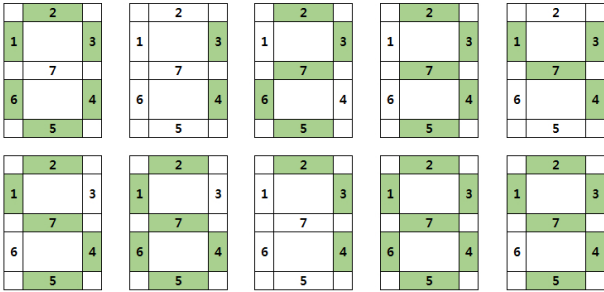
7 segment 12개를 사용하여 3자리수(A)와 3자리수(B)의 결과(C)를 6자리수로 표현하는 작업을 하려고 한다.

7 segment는 7개의 LED로 구성되어 있으며, 아래 그림에 LED 번호가 적혀 있고, LED의 ON, OFF 상태에 따라 0 ~ 9의 숫자를 표시할 수 있다. 예를 들어 2, 3, 4, 5, 7 번 LED가 ON인 경우 숫자 3이 된다.



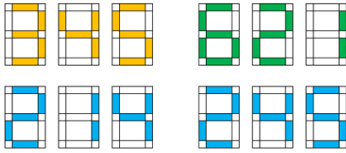
[7 segment 구성] [숫자 3 표시]

7 segment에 0 ~ 9의 숫자를 표시하면 다음과 같다. 아래 그림에서 색이 칠해져 있는 것이 LED ON, 칠해지지 않은 것이 LED OFF를 의미한다.

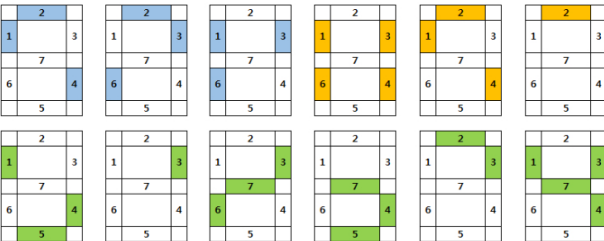


그런데, 7 segment의 LED가 고장이나 ON 상태로 표시되어야 할 부분들이 OFF 상태로 표시되는 것이 있다.

3개의 7 segment에 0-999의 숫자를 입력하고, 또 다른 3개의 7 segment에 0-999의 숫자를 입력했을 때, 아래의 6개의 7 segment에 결과가 표시된다. 다음은 345 * 621 = 214245가 표시된 예이다.



다음은 고장이나 LED가 OFF된 상태를 보여준다. OFF 되어 있는 LED는 고장이 난 것인지 숫자표시를 위해 신호를 받은 것인지 알 수 없는 상태이다. 즉, 1, 2, 4번 LED가 ON 된 경우는 0, 5, 6, 8, 9가 될 수 있고, 2, 3, 6번 LED가 ON 된 경우 0, 2, 8이 표시 될 수 있다.



위의 입력에서 6자리 7 segment에 표시 가능한 두 숫자의 곱 중 최소인 것은 찾으면 028 * 083 = 002324 가 된다.

LED의 상태를 입력 받아 6자리 7 segment에 표시 가능한 두 숫자의 곱 중 최소인 것을 찾아 출력하라.

```
#include
#define MAX (12+10)

char a[MAX][10];
int num[MAX][10]; // num[1][1] = 0 (사용가능) num[1][2] = 1 (불가능)
int data[10][10] = {
    { 0 },
    { 4, 1, 2, 3, 7 },
    { 2, 1, 4 },
    { 2, 5, 6 },
    { 1, 2 },
    { 3, 1, 4, 7 },
    { 6, 1, 3, 4, 5, 7, 9 },
    { 3, 1, 7, 0 } };

int n1[1000];
int n1Cnt, n2Cnt;
int n2[1000];

void input(void)
{
    int i;
    for (i = 1; i <= 12; i++)
        scanf("%s", &a[i][1]);
}

void guessNumber(void)
{
    }
```

```
int i, j, k;
for (i = 1; i <= 12; i++) // 12개 숫자
{
    for (j = 1; j <= 7; j++) // 7개 segments
    {
        if (a[i][j] == '0') continue;
        for (k = 1; k <= data[j][0]; k++)
        {
            num[i][data[j][k]] = 1; // 안되는 숫자에 1
        }
    }
}

int getNum(int base, int * arr)
{
    int i, j, k;
    int cnt = 0;
    int p1 = base + 1;
    int p2 = base + 2;

    for (i = 0; i <= 9; i++)
    {
        for (j = 0; j <= 9; j++)
        {
            for (k = 0; k <= 9; k++)
            {
                if (num[base][i] == 0 && num[p1][j] == 0 && num[p2][k] == 0)
                {
                    arr[++cnt] = i * 100 + j * 10 + k;
                }
            }
        }
    }
    return cnt;
}

int checkNum(int r)
{
    int i;

    for (i = 12; i >= 7; i--)
    {
        if (num[i][r % 10] == 1) return 0;
        r /= 10;
    }
    return 1;
}

int getSolution(void)
{
    int i, j;
    int r;
    int min = 0x7FFFFFFF;
    n1Cnt = getNum(1, n1);
    n2Cnt = getNum(4, n2);
    for (i = 1; i <= n1Cnt; i++)
    {
        for (j = 1; j <= n2Cnt; j++)
        {
            r = n1[i] * n2[j];
            if (min > r && checkNum(r) == 1) min = r;
        }
    }
    return min;
}

int main(void)
{
    input();
    guessNumber();
    printf("%d\n", getSolution());
    return 0;
}
```

입력 설명

하나의 행이 1개의 세그먼트에 대한 입력이며 12개 세그먼트의 입력이 A 3개, B 3개, C 6개 순서대로 입력된다.
1개의 행에는 7개의 숫자가 있으며 LED 번호 순서대로 ON/OFF의 상태가 빈칸 없이 입력된다.
1은 LED ON, 0은 LED OFF를 의미한다.

출력 설명

LED의 상태를 입력 받아 6자리 7 segment에 표시 가능한 두 숫자의 곱 중 최소인 것을 찾아 출력하라.

입력 예시

```
1101000
0110010
1010010
1011010
1101100
0100000
1001100
0010000
0010011
0001101
0110000
1011001
```

출력 예시

2324

부가정보

[제출]

시스템 운영 및 패스워드 초기화 요청 : keyseek@naver.com
(주)윌텍, (주)윌텍한기술, (주)기본기술, (주)임베디스, (주)에듀닉스, codexpert 연구소
Copyright © Willek Embedded System Experts Group