문제 풀기 전에 살짝 봤을 땐 너무 어려워 보여서 걱정했는데, 풀어보니 생각보다 어려운 문제는 아니다.
https://www.acmicpc.net/problem/1018

1018번: 체스판 다시 칠하기
문제 지민이는 자신의 저택에서 MN개의 단위 정사각형으로 나누어져 있…
www.acmicpc.net

체스판 다시 칠하기 성공

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	77681	37218	29954	48.009%

문제

지민이는 자신의 저택에서 MN개의 단위 정사각형으로 나누어져 있는 M×N 크기의 보드를 찾았다. 어떤 정사각형은 검은색으로 칠해져 있고, 나머지는 흰색으로 칠해져 있다. 지민이는 이 보드를 잘라서 8×8 크기의 체스판으로 만들려고 한다.

체스판은 검은색과 흰색이 번갈아서 칠해져 있어야 한다. 구체적으로, 각 칸이 검은색과 흰색 중 하나로 색칠되어 있고, 변을 공유하는 두 개의 사각형은 다른 색으로 칠해져 있어야 한다. 따라서 이 정의를 따르면 체스판을 색칠하는 경우는 두 가지뿐이다. 하나는 맨 왼쪽 위 칸이 흰색인 경우, 하나는 검은색인 경우이다.

보드가 체스판처럼 칠해져 있다는 보장이 없어서, 지민이는 8×8 크기의 체스판으로 잘라낸 후에 몇 개의 정사각형을 다시 칠해야겠다고 생각했다. 당연히 8*8 크기는 아무데서나 골라도 된다. 지민이가 다시 칠해야 하는 정사각형의 최소 개수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N과 M이 주어진다. N과 M은 8보다 크거나 같고, 50보다 작거나 같은 자연수이다. 둘째 줄부터 N개의 줄에는 보드의 각 행의 상태가 주어진다. B는 검은색이며, W는 흰색이다.

출력

첫째 줄에 지민이가 다시 칠해야 하는 정사각형 개수의 최솟값을 출력한다.

예제 입력 1 복사

```
8 8
WBWBWBWB
BWBWBWBW
WBWBWBWB
BWBBBWBW
WBWBWBWB
BWBWBWBW
WBWBWBWB
BWBWBWBW
```

예제 출력 1 복사

```
1
```

예제 입력 2 복사

```
10 13
BBBBBBBBWBWBW
BBBBBBBBBWBWB
BBBBBBBBWBWBW
BBBBBBBBBWBWB
BBBBBBBBWBWBW
BBBBBBBBBWBWB
BBBBBBBBWBWBW
BBBBBBBBBWBWB
WWWWWWWWWWBWB
WWWWWWWWWWBWB
```

예제 출력 2 복사

```
12
```

예제 입력 3 복사

```
8 8
BWBWBWBW
WBWBWBWB
BWBWBWBW
WBWBWBWB
BWBWBWBW
WBWBWBWB
BWBWBWBW
WBWBWBWB
```

예제 출력 3 복사

```
0
```

예제 입력 4 복사

```
9 23
BBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBW
```

예제 출력 4 복사

```
31
```

예제 입력 5 복사

```
10 10
BBBBBBBBBB
```

예제 출력 5 복사

```
0
```

```
BBWBWBWBWB
BWBWBWBWBB
BBWBWBWBWB
BWBWBWBWBB
BBWBWBWBWB
BWBWBWBWBB
BBWBWBWBWB
BWBWBWBWBB
BBBBBBBBBB
```

예제 입력 6 복사

```
8 8
WBWBWBWB
BWBWBWBW
WBWBWBWB
BWBBBWBW
WBWBWBWB
BWBWBWBW
WBWBWWWB
BWBWBWBW
```

예제 출력 6 복사

```
2
```

예제 입력 7 복사

```
11 12
BWWBWWBWWBWW
BWWBWBBWWBWW
WBWWBWBBWWBW
BWWBWBBWWBWW
WBWWBWBBWWBW
BWWBWBBWWBWW
WBWWBWBBWWBW
BWWBWBWWWBWW
WBWWBWBBWWBW
BWWBWBBWWBWW
WBWWBWBBWWBW
```

예제 출력 7 복사

```
15
```

출처

- 문제를 번역한 사람: baekjoon
- 데이터를 추가한 사람: jh05013
- 문제를 다시 작성한 사람: jh05013

알고리즘 분류

- 브루트포스 알고리즘

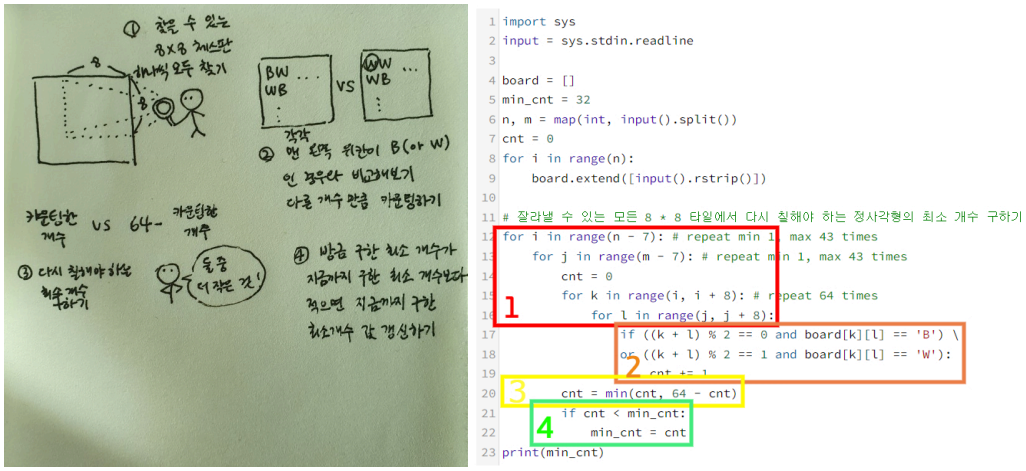
메모

```python
# 제출 번호 49748116번 소스 코드, 아이디 panpan8, 언어 Python 3
import sys
input = sys.stdin.readline

board = []
min_cnt = 32
n, m = map(int, input().split())
cnt = 0
for i in range(n):
    board.extend([input().rstrip()])

# 잘라낼 수 있는 모든 8 * 8 타일에서 다시 칠해야 하는 정사각형의 최소 개수 구하기
for i in range(n - 7): # repeat min 1, max 43 times
    for j in range(m - 7): # repeat min 1, max 43 times
        cnt = 0
        for k in range(i, i + 8): # repeat 64 times
            for l in range(j, j + 8):
                if ((k + l) % 2 == 0 and board[k][l] == 'B') \
                or ((k + l) % 2 == 1 and board[k][l] == 'W'):
                    cnt += 1
        cnt = min(cnt, 64 - cnt)
        if cnt < min_cnt:
            min_cnt = cnt
print(min_cnt)
```

코드를 그림과 함께 살펴보자.
그림과 같은 번호로 묶인 코드를 함께 살펴보면 좀 더 쉽게 이해하는 데 도움이 될 것이다.



우선 입력을 받아야 한다.
위 사진 1~2번째 줄처럼 알고리즘 문제 풀 때 input()을 readline()으로 선언하고 시작하면 좋다.
readline()은 input()과 같은 입력 함수지만, 속도가 더 빠르기 때문이다.

이렇게 적어놓고 평소처럼 input 쓰듯이 코드를 작성하면 되는데, 대신 주의해야 할 점이 하나 있다.
바로 readline()은 개행문자까지 입력을 받는다는 것.
그래서 우측 개행문자를 제거하기 위해 9번째 줄에서 rstrip()을 사용한 것이다.

5번째 줄의 min_cnt는 출력할 최솟값을 담을 변수인데, 이 값을 32로 설정한 이유는 뒤에서 설명할 예정이다.
7번째 줄의 cnt는 사실 14번째 줄에서 선언 및 초기화가 이루어질 수 있기 때문에 지워도 문제없다.

1
입력이 끝났다면, 입력받은 보드를 잘라 만들 수 있는 8 * 8 체스판을 모두 찾아야 한다.
문제처럼 보드의 가로를 m, 세로가 n일 때 만들 수 있는 8 * 8 체스판의 개수는 (m - 7) * (n - 7)개이다.

12, 13번째 줄 range()의 인자값인 n - 7, m - 7이 바로 이것과 관련이 있다.
n과 m은 8 이상 50 이하인 자연수이므로, 보드로 최대 43 * 43개의 체스판을 만들 수 있다.

2
이제 모든 만들 수 있는 체스판을 순회하며 맨 왼쪽 위 칸이 B인 보드와 비교하여 다른 색상의 정사각형 개수를 셀 것이다.
사실 내 코드에서는 같은 색상의 정사각형 개수를 셌다.
맨 왼쪽 위 칸이 B인 보드와 비교했을 때 같은 색상인 정사각형 개수와 다른 색상인 정사각형 개수의 합은 64로 같으므로, 어차피 같은 색상의 정사각형 개수를 세더라도 64에서 방금 센 개수를 빼면 다른 색상 정사각형 개수의 합을 알 수 있기 때문이다.

체스판의 조건을 만족하려면 보드의 가로, 세로 인덱스의 값을 합한 값이 홀수인 타일과 짝수인 타일은 각각 다른 색이어야 한다.
정말 그런지 확인해 보고 싶다면 아래 그림을 인쇄해 두 인덱스의 합이 홀수인 사각형과 짝수인 사각형을 각각 다른 색으로 색칠해 보자.
체스판의 형태가 나타날 것이다.

나는 합이 홀수인 타일은 W, 짝수인 타일은 B인 사각형 개수를 셌다.
8 * 8개의 사각형의 색상을 확인하므로, 각 체스판마다 총 64번의 연산이 필요하다.

여기서 시간 제한 조건을 한 번 짚고 넘어가자.
문제의 시간 제한은 2초이고 파이썬은 1초에 대략 2천만 번 연산이 가능하므로, 입출력 등에서 소요되는 시간을 배제하고 생각해 본다면 대략 4천만 번 이하의 연산이 이루어져야 한다.
가장 많은 연산이 이루어지는 최악의 경우는 n과 m이 50일 때이다.
만들 수 있는 모든 체스판에서 타일 색상을 비교하는 데 43 * 43 * 8 * 8 = 118,336번의 연산 횟수가 필요하다.
4천만에 비해 크게 작은 숫자이니 시간 초과는 이제 크게 걱정하지 않아도 된다.

board[N][M]

board[0][0]	board[0][1]	board[0][2]	board[0][3]	board[0][4]	board[0][5]	board[0][6]	board[0][7]
board[1][0]	board[1][1]	board[1][2]	board[1][3]	board[1][4]	board[1][5]	board[1][6]	board[1][7]
board[2][0]	board[2][1]	board[2][2]	board[2][3]	board[2][4]	board[2][5]	board[2][6]	board[2][7]
board[3][0]	board[3][1]	board[3][2]	board[3][3]	board[3][4]	board[3][5]	board[3][6]	board[3][7]
board[4][0]	board[4][1]	board[4][2]	board[4][3]	board[4][4]	board[4][5]	board[4][6]	board[4][7]
board[5][0]	board[5][1]	board[5][2]	board[5][3]	board[5][4]	board[5][5]	board[5][6]	board[5][7]
board[6][0]	board[6][1]	board[6][2]	board[6][3]	board[6][4]	board[6][5]	board[6][6]	board[6][7]
board[7][0]	board[7][1]	board[7][2]	board[7][3]	board[7][4]	board[7][5]	board[7][6]	board[7][7]

3
우리는 다시 칠해야 하는 최소 사각형의 개수를 알아야 하므로, 왼쪽 위 칸이 흰색 또는 검은색인 체스판과 비교했을 때 다른 색상인 사각형의 개수를 구하고 그중 더 적은 쪽을 선택해야 한다.
앞서 말했듯이, 체스판을 색칠하는 경우 하나와 비교했을 때 같은 색상인 정사각형 개수와 다른 색상인 정사각형 개수의 합은 64로 같다.

아래 두 가지 식이 성립한다.
왼쪽 위 칸이 검은색인 체스판과 비교했을 때 같은 색상인 정사각형 개수 = 오든쪽 위 칸이 흰 색인 체스판과 비교했을 때 같은 색상인 정사각형 개수
왼쪽 위 칸이 검은색인 체스판과 비교했을 때 다른 색상인 정사각형 개수 = 오든쪽 위 칸이 흰 색인 체스판과 비교했을 때 다른 색상인 정사각형 개수

말로 풀어쓰니 복잡한데, 쉽게 말해 방금 센 정사각형의 개수와 64에서 그 값을 뺀 것 중 더 작은 쪽을 선택하면 된다는 말이다.
이를 위해 인자 중 더 작은 값을 리턴하는 min()을 사용하고, 리턴 값은 cnt에 저장한다.

4
min_cnt와 cnt 중 더 작은 쪽을 선택하여 min_cnt에 저장한다.
이제 와서 보니 if문을 쓰지 않고 min()을 써도 될 것 같다.
아예 min(cnt, 64 -cnt, min_cnt)로 3, 4번을 합쳐도 괜찮겠다.

이 과정을 만들 수 있는 모든 체스판 개수만큼 반복하여 진행하면, 다시 칠해야 하는 사각형 개수의 최솟값을 구할 수 있다.

n차원 배열 연산하다 인덱싱할 때 인덱스 값을 헷갈릴 때가 있다.
헷갈릴 때마다 인덱스나 배열 값을 출력해 보는 것도 방법이지만, 그냥 변화 형태를 외워두는 것이 편하다.

2차원 배열의 경우 이렇게 변한다.
x(가로), y(세로) → 배열[y][x]

여기서 z축이 추가되어 3차원 배열이 된다면?
x, y, z → 배열[z][y][x]