

문제	3주차 - A																																																																
제목	흑백 이미지 처리																																																																
내용	<p>인하는 새로운 이미지 압축 알고리즘을 개발하기 위해 <그림 1>과 같은 흑백 이미지에 대한 처리방법을 생각해 냈다. 인하가 생각한 방법은 다음과 같다.</p> <div><table><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table></div> <p><그림 1></p> <ol style="list-style-type: none">1. 각 픽셀마다 하얀색 픽셀은 0, 검은색 픽셀은 1로 나타난 정사각형의 흑백 이미지를 입력으로 받는다. 이때 정사각형의 크기는 $N \times N$ ($N = 2^x, 1 \leq x \leq 8$) 이다. 그림 1의 경우 8×8 이미지이다.2. 이미지가 단일 색이 아닐 경우 가로와 세로를 각각 2등분하여 <그림 2>의 I, II, III, IV와 같이 4개의 $N/2 \times N/2$ 블록으로 나눈다. 단, 이미지가 단일 색상일 경우 한 개의 블록을 구성한다.3. 나누어진 I, II, III, IV 각각의 블록에 대해 한가지 색이 아닌 픽셀을 갖는 블록에 대해서는 2와 같은 방법으로 4개의 블록으로 나눈다.4. 3의 과정을 각각의 모든 블록이 한가지 색을 가질 때까지 반복한다. <div></div> <p><그림 2></p> <p>이미지의 한 변의 길이 N 과 각 픽셀의 색상 값(하얀색 0, 검정색 1)이 주어질 때, 위와 같은 방법으로 블록을 최소한으로 생성할 경우 하얀색 블록과 검은색 블록이 각각 몇 개가 될지 계산하는 프로그램을 작성하시오.</p> <p>사용할 수 있는 언어는 C, C++로 제한한다. 프로그램의 실행 시간은 5초를 초과할 수 없다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin 입력 속도를 개선할 수 있다.</p> <pre>std::ios::sync_with_stdio(false);</pre>	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	1	1	1	0	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1
1	1	0	0	0	0	1	1																																																										
1	1	0	0	0	0	1	1																																																										
0	0	0	0	1	1	0	0																																																										
0	0	0	0	1	1	0	0																																																										
1	0	0	0	1	1	1	1																																																										
0	1	0	0	1	1	1	1																																																										
0	0	1	1	1	1	1	1																																																										
0	0	1	1	1	1	1	1																																																										

입력 형식		<p>입력은 standard in으로 다음과 같이 주어진다.</p> <ol style="list-style-type: none"> 1. 첫 번째 줄에는 테스트케이스의 수 T가 주어진다. ($0 < T \leq 500$) 2. 두 번째 줄에는 입력으로 주어지는 흑백 이미지의 한 변의 길이 N이 주어진다. ($2^1 \leq N \leq 2^8$) 3. 세 번째 줄부터 $2 + N$까지의 N개의 줄에는 흑백 이미지의 각 가로줄의 정사각형 칸들이 윗줄부터 차례로 주어진다. 각 줄은 N개의 숫자이며, 하얀색 픽셀은 0, 검은색 픽셀은 1 값이 빈칸을 사이에 두고 하나씩 주어진다. 4. 그 이후에는 위의 2, 3 단계가 $T - 1$ 번 반복된다.
출력 형식		<p>출력은 standard out으로 표시하며, 총 T줄로 이루어진다. 각 줄에는 각 테스트케이스 별로 하얀색 블록 수와 검은색 블록 수를 빈칸을 사이에 두고 출력한다.</p>
예	입력	<pre> 2 8 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 4 1 1 0 0 1 1 0 0 0 0 1 1 0 0 1 0 (빈 줄) </pre>
	출력	<pre> 9 7 3 4 (빈 줄) </pre>