1. 공격하는 입장.

내가 내 돌이 4개 혹은 5개여서 6목을 만들 수 있는지 check 하는 함수

Checkmine (int s[19][19], char state, int& pos1, int& pos2, int p, int q)

를 부르면됩니다.

Checkmineblank함수는 checkmine 함수안에서 부르므로 신경쓰지 않으셔도 됩니다.

우선 s는 보드판입니다. I, k는 row, column기준이며 s[i][k] = 0이면 빈칸, 1이면 흑색, 2이면 백색돌이 놓여있다고 가정하였습니다.

State는 나의 색깔입니다. 즉 흑색인지, 백색인지를 넣어주시면 됩니다. 내가 흑돌이라면 1을, 백돌이라면 2를 넣어주시면 됩니다.

Int &pos1, int &pos2는 int 변수의 주소값을 가져간다는 의미로,

그냥 신경쓰실 필요 없이

Int pos1, pos2를 선언하신 뒤

Checkming(s, state, pos1, pos2, p, q)와 같은 형식으로 넣어주시면 됩니다. (포인터로 넣으시면 안되고 그냥 변수명을 그대로 넣으시면 됩니다.)

함수가 끝난 뒤에 찾은 index가 있다면 index가 해당 변수들(pos1, pos2)에 들어가 있을 것이고, 그렇지 않다면 -1이 들어가 있을 것입니다.

p, q는 내가 직전 상태에서 넣었던 돌의 위치입니다. (상대방 돌의 위치가 아니라 내 돌의 위치!!) 그러므로 내가 직전에 놓은 돌이 무엇이었는지 그 위치를 가지고 있어야 합니다.

S[i][k]에 넣었었다면 p = I, q = k 가 되는 것입니다.

주의하셔야 할 점은 직전에 놓은 돌이 두 번 이므로, 각 점에 대해 이 함수를 각각 불러주셔야 합니다. 예를들어 직전에 (1, 3)과 (2, 4)에 놓았다면 p=1, q=3으로 checkmine함수를 한 번 부르고, 만약 불렀을 때 pos1, pos2값이 둘 다 -1이었다면, p=2, p=4로 checkmine을 한 번 더 불러주셔야 합니다. 만약 p=1, q=3일 때 이미 pos1, pos2값이 -1이 아니라 index를 찾았다면 굳이 (2, 4)에 대해서 부를 필요는 없지만, (1,3)으로 index를 찾지 못했다 하더라도 (2,4)로는 찾을 수도 있기 때문에 주의해서 함수를 쓰셔야 할 것 같습니다.

2. 방어하는 입장

상대방 돌이 4개 혹은 5개여서 6목을 만들 수 있는지 check 하는 함수

checkopponent(int s[19][19], char state, int& pos1, int& pos2, int p, int q)

함수를 부르면됩니다.

Checkoppenentblank함수는 checkoppenent 함수안에서 부르므로 신경쓰지 않으셔도 됩니다.

우선 s는 보드판입니다. I, k는 row, column기준이며 s[i][k] = 0이면 빈칸, 1이면 흑색, 2이면 백색돌이 놓여있다고 가정하였습니다.

State는 **상대방**의 색깔입니다. 즉 흑색인지, 백색인지를 넣어주시면 됩니다. 내가 백돌 = 상대방 흑돌 = state에는 1. 내가 흑돌 = 상대방 백돌 = state는 2

Int &pos1, int &pos2는 int 변수의 주소값을 가져간다는 의미로,

그냥 신경쓰실 필요 없이

Int pos1, pos2를 선언하신 뒤

Checkming(s, state, pos1, pos2, p, q)와 같은 형식으로 넣어주시면 됩니다. (포인터로 넣으시면 안되고 그냥 변수명을 그대로 넣으시면 됩니다.)

함수가 끝난 뒤에 찾은 index가 있다면 index가 해당 변수들(pos1, pos2)에 들어가 있을 것이고, 그렇지 않다면 -1이 들어가 있을 것입니다.

p, q는 내 턴이 돌아오기 직전 상대방이 둔 돌의 위치입니다.

상대방이 S[i][k]에 넣었었다면 p = I, q = k 가 되는 것입니다.

주의하셔야 할 점은 상대방이 직전에 놓은 돌이 두 번 이므로, 이 함수 역시 각 점에 대해 이 함수를 각각 불러주셔야 합니다. 예를들어 직전에 상대방이 (1, 3)과 (2, 4)에 놓았다면 p=1, q=3으로 checkmine함수를 한 번 부르고, 만약 불렀을 때 pos1, pos2값이 둘 다 -1이었다면, p=2, p=4로 checkmine을 한 번 더 불러주셔야 합니다. 만약 p=1, q=3일 때 이미 pos1, pos2값이 -1이 아니라 index를 찾았다면 굳이 (2, 4)에 대해서 부를 필요는 없지만, (1,3)으로 index를 찾지 못했다 하더라도 (2,4)로는 찾을 수도 있기 때문에 주의해서 함수를 쓰셔야 할 것 같습니다.

\*\* 위험한 위치가 여러 곳 존재한다 하더라도, 이 함수를 통해서는 단 한곳만 방어할 수 있습니다. 이 함수가 알려주는 index를 막지 않으면 다음 턴에 지겠지만, 이 함수가 알려주는 index를 막는다고 해서 100% 안전하진 않습니다. (다른 곳이 위험할 수 있으므로) 참고 부탁드립니다.