1. Submit a documented source code, and, if relevant, executables. This should include the source for all eight programs

   Following are eight source code you will find the zip:

   MiniMaxOpening.py

   MiniMaxGame.py

   ABGame.py

   ABOpening.py

   MiniMaxGameBlack.py

   MiniMaxOpeningBlack.py

   MiniMaxOpening.py

   MiniMaxGameImproved.py

2. Show examples of the program output when applied to several positions. Give at least two cases in which alpha-beta produces savings over MINIMAX.

   Input: WxWxWxxxWxxxBBWBBx
   Depth: 2
   Step: Opening
   Number of positions Evaluated by MiniMax: 564
   Number of positions evaluated by ABOpening: 221

   Input: WxWxWxxxWxxxBBWBBx
   Depth: 2
   Step: Game
   Number of positions Evaluated by MiniMax: 18
   Number of positions evaluated by ABOpening: 32

3. Show at least two examples where your evaluation function produced different moves than the standard evaluation function. Write a short (one or two paragraphs) explanation of why you believe your function to be an improvement over the function proposed by the instructor.

   Change made to the static evaluation when number of white and Black are greater than 3

   And all steps for black are not 0 then the improved evaluation function would calculate the number of closemill of white – closemill of black which tries to maximize the opportunity to again close the same mill and get black out.

For ex1: consider the input WBWxWBxxWxxxxBWBBx

In this input white has made a closemill and would not be able to make any other mill in next few steps the only mill it can close is (0,2,4), so the improved version would try to get to move W such that the black does not close a mill and White can close mill again

Output using Improved: WBxWWBxxWxxxxBWBBx

You can see that 3rd (2) W has moved and will again close the mill in next round, now black cannot come in that place.

Ouput using MiniMax: WBWxWBxxWxxxxBxBBW

As compared the static function given in the handout the one I put forth is much riskier but can make bold choices.

Ex2: Input: BWWWxxWWWxxBxxxBBx

On this example the output by improved function is BWWWxxWxWxxBxxWBBx which make

Sensible move to move f3(8) as W at position 7 if moved can cause black to move

Now the handout function gives BWWWxWxWWxxBxxxxBx move the white at e3(7) and here black can come in from e4 can occupy e3 hence its too risky.

Both example shows us that the improved function perform better.