Programming Problem 3:

The heuristic function that we used is:

h(n) -> number of misplaced tiles

We evaluate the heuristic function for each state at every level.

The algorithm we tried to implement for the problem 3 is as follows:

Begin:

1) Open_States   =   Initial State

2) Closed   =   []

3) While ([Open_State] is not null) :

i) Remove the leftmost state from Open_States; Let's call it X

ii) Evaluate X

iii) If (X is Goal State)

a) Check other elements in the Open_States list and evaluate them.

b) Return the path to X
iv) else

a) Generate Children of X

b) For each child of X, do the following based on the case

I) Case : (Child not in Open_States or Closed lists)
Find the heuristic value of the child and add child to Open

II) Case : (Child is already in Open_States list but not in Closed)
If the child was reached by shorter path, then give the
state on Open_States the shorter path

III) Case: (Child is in Closed list)
If the child was reached by a shorter path, then remove the
child from Closed and add it to Open

v) Put X into Closed list

vi) Re-order the list Open_State by sorting based on the heuristic function

Following is our goal state:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$