LAPORAN TUGAS KECIL 3 IF2211 STRATEGI ALGORITMA

PENYELESAIAN PERSOALAN 15-PUZZLE DENGAN ALGORITMA BRANCH AND BOUND

PUTRI NURHALIZA

13520066

TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

SEMESTER II 2021/202

Daftar Isi

A. Cara Kerja Program *Branch and Bound* Pada 15-Puzzle

1. Cara Kerja Program

   Program disusun oleh *Priority Queue* untuk menyimpan simpul yang diurutkan berdasarkan *cost* terkecil. Kelas *node* merupakan suatu kelas yang menyimpan informasi simpul tersebut, seperti matriks *puzzle*, posisi ubin kosong, *cost*, kedalaman, *parent node*, dan pergerakan yang dilakukan untuk mencapai node tersebut dari *parent* nya. Jika *goal state* dapat dicapai dari initial state (*reachable*), maka pohon akan dibangun secara dinamis. Dimulai dengan memasukkan root (initial state) ke dalam *priority queue*. Lalu, dilakukan algoritma yang dijelaskan pada poin 2 di bawah. Algoritma ini akan menghasilkan 1 simpul solusi. Kemudian, program utama akan memanggil fungsi printRute untuk mencetak setiap matriks dari posisi awal ke posisi akhir di lintasan yang sesuai dengan memanfaatkan rekursivitas dari *parent node* yang telah disimpan.

2. Algoritma

   a. Masukkan simpul akar ke dalam antrian PQ (*priority queue*). Prioritas yang digunakan pada *priority queue* adalah simpul yang memiliki *cost* terkecil. Simpul akar adalah puzzle dalam keadaan awal.
   b. Jika PQ kosong, berhenti.
   c. Jika PQ tidak kosong, dari antrian PQ, pilih simpul i yang mempunyai nilai *cost* paling kecil (dengan *priority queue*, maka simpul ini berada di paling depan).
   d. Jika simpul i tersebut adalah simpul solusi (*goal node*, sesuai dengan keadaan *final puzzle* yang diinginkan), berarti solusi sudah ditemukan dan berhenti.
   e. Jika simpul i bukan simpul solusi, maka bangkitkan semua anak-anaknya. Anak-anak dalam permasalahan ini adalah *puzzle* hasil pergerakan ubin kosong yang memungkinkan baik ke kiri, kanan, atas, atau bawah. Jika simpul i tidak mempunyai anak, kembali ke langkah b.
   f. Untuk setiap anak j dari simpul i, hitung *cost*-nya dan masukkan semua anak-anak tersebut ke dalam PQ.
   g. Kembali ke langkah b.

3. Perhitungan cost

   Cost setiap simpul P dihitung dari panjang lintasan dari simpul akar ke simpul P (level kedalaman) yang disebut $f(P)$ ditambah dengan panjang lintasan terpendek dari simpul P ke simpul solusi yang ditaksir dengan jumlah ubin yang tidak kosong yang tidak sesuai dengan keadaan final yang disebut $\hat{g}(P)$. Singkatnya,

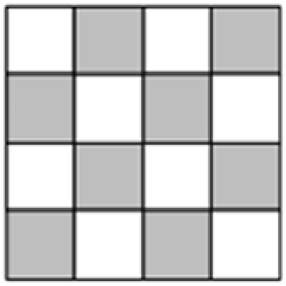
   $$c(P) = f(P) + \hat{g}(P)$$

   Pada program ini, jika terdapat lebih dari 1 simpul dnegan cost yang sama, maka yang menjadi prioritas adalah simpul yang level nya lebih dalam.

4. Heuristics

   a. Tidak semua keadaan awal sembarang dapat mencapai keadaan final. Sebelum dilakukan pencarian, akan diperiksa terlebih dahulu apakah keadaan akhir dapat dicapai dari keadaan awal *puzzle*. Dicek dengan teorema bahwa keadaan final hanya dapat dicapai jika $\sum_{i=1}^{16} KURANG(i) + X$. bernilai genap. $KURANG(i)$ merupakan suatu fungsi untuk menghitung banyaknya ubin bernomor j dengan $j < i$ namun

$posisi(j) > posisi\ (i)$. Sementara itu, X akan bernilai 1 jika berada di daerah yang diarsir, dan bernilai 0 jika sebaliknya.



Gambar 1.1. Ilustrasi arsiran ubin untuk menentukan nilai X

b. Membatasi pergerakan ubin kosong untuk kembali ke keadaan puzzle yang sudah pernah ada sebelumnya. Misal, jika sebelumnya ubin kosong bergerak kebawah, maka pada langkah selanjutnya ubin kosong tidak boleh kembali ke atas. Pada program ini, juga diperiksa apakah suatu matriks pernah dibangkitkan sebelumnya. Jika ya, maka simpul tersebut tidak akan diangkitkan lagi. Hal ini bertujuan agar tidak terjadi perulangan untuk mengoptimasi pencarian.

B. *Source Code* Program (Bahasa Pemograman Python)

Program dibuat secara modular yang dikelompokkan ke dalam 5 file.

1. inputPuzzle.py
   Terdiri dari dua fungsi untuk mendapatkan *initial state* puzzle, dari file dan dari randomizer untuk me-*return* matriks.

```python
import random
import numpy as np

def inputFromFile(filename):
    dir = "../test/" + filename
    file = open(dir, "r")
    content =  (open(dir).read().split())
    file.close()
    puzzle = np.zeros((4,4), dtype=int)
    for i in range(0, 16):
        puzzle[i // 4, i % 4] = int(content[i])
    return np.array(puzzle)

def inputFromRandom():
    tmp = random.sample(range(1, 17), 16)
    puzzle = np.zeros((4,4), dtype=int)
    for i in range(0, 16):
        puzzle[i // 4, i % 4] = tmp[i]
    return np.array(puzzle)
```

2. node.py
   Class node, menyimpan informasi dari suatu node

```python
# Class node, store information of a puzzle node
class node:
    def __init__(self, matrix, emptyTilePos, cost, level, parent, prevMove):
        self.matrix = matrix                    # matrix representation of the puzzle
        self.emptyTilePos = emptyTilePos        # position of empty tile
        self.cost = cost                        # c(i) = f(i) + g(i)
        self.level = level                      # depth from root
        self.parent = parent                    # node of parent
        self.prevMove = prevMove                # previous move of empty tile to reach
this node

    # set priority when enqueue a node to the priority queue based on cost
    def __lt__(self, next):
        if self.cost == next.cost:
            return self.level > next.level
        return self.cost < next.cost
```

3. helper.py
   Terdiri dari fungsi-fungsi untuk membantu algoritma branch and bound dan program utama

```python
n = 4       # Puzzle size is n x n

# Function to print matrix
def printmatrix(matrix):
    for i in range(n):
        print("-----" * n)
        for j in range(n):
            print("|", end=" ")
            if matrix[i][j] == n**2:
                print("  ", end=" ")
            else:
                print(str(matrix[i][j]).ljust(2), end = " ")
        print("|")
    print("-----" * n)

# Recursive function to print the solution from root
def printRute(node):
    if node == None:
        return
    printRute(node.parent)
    if node.parent == None:
        print(f"LEVEL: {node.level}, INITIAL")
    else:
        print(f"LEVEL: {node.level}, PREVIOUS MOVE: {node.prevMove}")
    printmatrix(node.matrix)
    print()

# Return tuple (i, j) of position of x in matrix
def getXPos2D(matrix, x):
    for i in range(n):
        for j in range(n):
            if matrix[i][j] == x:
                return (i, j)

# Return posion of x in matrix (0-15)
def getXPos(matrix, x) -> int:
    x1, x2 = getXPos2D(matrix, x)
    return x1 * n + x2

# Count tile with number j which i > j but position(j)<position(i)
def kurangI(matrix, i) -> int:
    kurangI = 0
    for j in range (1, i):
        if getXPos(matrix, j) > getXPos(matrix, i):
            kurangI += 1
    return kurangI

# Sum of kurangI(i) + x for i from 1 to n**2
def SumKurangIplusX(matrix) -> int:
    SumKurangI = 0
    for i in range(1, n**2+1):
        SumKurangI += kurangI(matrix, i)
    x1, y1 = getXPos2D(matrix, n**2)    # empty tile position
    X = (x1 + y1) % 2                   # 1 if empty tile are in shadow area, 0 if not
    return SumKurangI + X

# Return true if matrix is solvable
def isReachable(matrix) -> bool:
    return SumKurangIplusX(matrix) % 2 == 0
```

4. solver.py
   Berisi algoritma utama serta fungsi perantara untuk menyelesaikan 15-Puzzle dengan algoritma *branch and bound*

```
import numpy as np
import copy

from inputPuzzle import *
from queue import PriorityQueue
from node import *
from helper import *

# Store the goal state of puzzle
goal = np.array([[ 1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11 , 12],
                [13, 14, 15, 16]])

# Store the name of direction of each move
direction = ['UP', 'RIGHT', 'DOWN', 'LEFT']


# function to create new node
def newNode(liveNode, newEmptyTilePos, prevMove, raisedMatrix) -> node:
    newMatrix = copy.deepcopy(liveNode.matrix)

    x1, y1 = liveNode.emptyTilePos
    x2, y2 = newEmptyTilePos

    # change position of empty tile to the other tile based on previous move
    newMatrix[x1][y1], newMatrix[x2][y2] = newMatrix[x2][y2], newMatrix[x1][y1]

    level = liveNode.level + 1                    # f(i)
    missplaced = calculateMissplaced(newMatrix)    # g(i)
    cost = level + missplaced                       # c(i) = f(i) + g(i)

    if (newMatrix.tobytes() in raisedMatrix):
        # if matrix has been raised before, it wont be raised again
        return None
    else:
        # raised the new node
        raisedMatrix[newMatrix.tobytes()] = True
        newnode = node(newMatrix, newEmptyTilePos, cost, level, liveNode, prevMove)
        return newnode

# count how many missplaced tile in matrix
def calculateMissplaced(matrix) -> int:
    count = 0
    for i in range(n):
        for j in range(n):
            if (matrix[i][j] != goal[i][j]):
                if matrix[i][j] == n**2:
                    continue
                count += 1
    return count

# move empty tile to the position based on direction
def move(currentPos, i):
    # i = 0: move up
    # i = 1: move right
    # i = 2: move down
    # i = 3: move left
    moverow = [-1, 0, 1, 0]
    movecol = [0, 1, 0, -1]

    x1, y1 = currentPos
    return [x1 + moverow[i], y1 + movecol[i]]

# Return true if this position are in valid in matrix
def isValid(position) -> bool:
    return position[0] >= 0 and position[0] < n and position[1] >= 0 and position[1] < n

# main algorithm to solve 15 puzzle based on Branch and Bound algorithm
def solve(initial):
    # Store the matrix that have been raised, unique
    raisedMatrix = {}



    pq = PriorityQueue()
    totalNodes = 0
    level = 0
    cost = calculateMissplaced(initial) + level
    emptyTilePos = getXPos2D(initial, n**2)
    root = node(initial, emptyTilePos, cost, level, None, None)
    pq.put(root)

    # Loop until the prioqueue is empty or the goal is found
    while not pq.empty():
        # get the node with lowest cost
        liveNode = pq.get()

        if liveNode.cost == liveNode.level:
            # g = 0, no more missplaced tiles, goal found
            return liveNode, totalNodes

        for i in range(4):
            # move empty tile to the position based on 4 possible direction, and raised
the new nodes
            newEmptyTilePos = move(liveNode.emptyTilePos, i)
            if isValid(newEmptyTilePos):
                child = newNode(liveNode, newEmptyTilePos, direction[i], raisedMatrix)
                if child != None:
                    pq.put(child)
                    totalNodes += 1
```

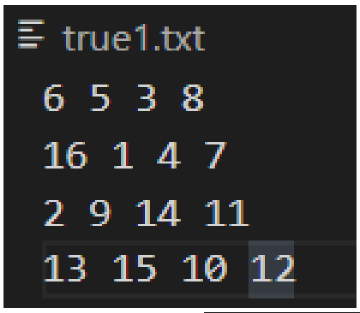
5. main.py
   File utama untuk mengatur flow program dan memanfaatkan fungsi-fungsi dari file-file di atas untuk mendapatkan input dan mencetak output yang sesuai.

```
import time
import os
from inputPuzzle import *
from node import *
from helper import *
from solver import *


print("-----------------WELCOME TO 15 PUZZLE SOLVER---------------------")
print("""

 ██╗███████╗    ██████╗ ██╗   ██╗███████╗███████╗██╗     ███████╗
███║██╔════╝    ██╔══██╗██║   ██║╚══███╔╝╚══███╔╝██║     ██╔════╝
╚██║███████╗    ██████╔╝██║   ██║  ███╔╝   ███╔╝ ██║     █████╗
 ██║╚════██║    ██╔═══╝ ██║   ██║ ███╔╝   ███╔╝  ██║     ██╔══╝
 ██║███████║    ██║     ╚██████╔╝███████╗███████╗███████╗███████╗
 ╚═╝╚══════╝    ╚═╝      ╚═════╝ ╚══════╝╚══════╝╚══════╝╚══════╝


""")
```

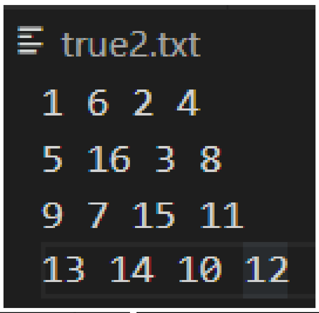
```
n = 4           # puzzle size is n x n
isRun = True

while(isRun):
    print("You can use initial puzzle from")
    print("1. File")
    print("2. Randomizer")
    opsi = int(input("Choose initial puzzle from: (1-2) \n>> "))
    inputExist = False

    if opsi == 1:
        filename = input("Enter filename: \n>> ")
        if (os.path.exists("../test/" + filename)):
            initial = inputFromFile(filename)
            inputExist = True
        else:
            print("File not found")
    elif opsi == 2:
        initial = inputFromRandom()
        inputExist = True
    else:
        print("Invalid input")

    print()
    if (inputExist):
        print("Initial Puzzle:")
        printmatrix(initial)
        print()
        print("Nilai Kurang(i)")
        for i in range(1, n**2+1):
            print(f"Kurang({i}) = {kurangI(initial, i)}")
        print(f"SumKurangI+X = {SumKurangIplusX(initial)}")
        print()

        if (isReachable(initial)):
            print("THE GOAL IS REACHABLE FROM THIS PUZZLE :D\n")
            print("Please wait for a while...\n")
            start = time.time()
            solution, totalNodes = solve(initial)
            end = time.time()
            printRute(solution)
            print("---------------------------------------------------")
            print(f"Time elapsed        : {end-start} seconds")
            print(f"Total nodes raised  : {totalNodes}")
        else:
            print("THE GOAL IS NOT REACHABLE FROM THIS PUZZLE :(")
        print()
```

```
loop = input("Wanna try another puzzle? (y/n) \n>> ")
if (loop.upper() == "Y"):
    isRun = True
    print()
else:
    print("--------------------THANK YOU--------------------")
    isRun = False
```

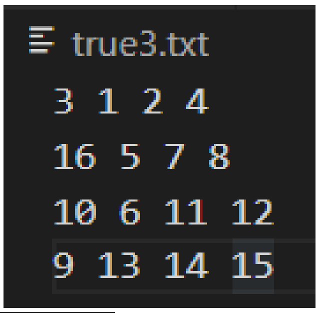
C. Contoh Instantiasi 15 Puzzle



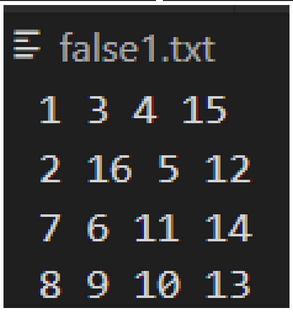
```
true1.txt
6 5 3 8
16 1 4 7
2 9 14 11
13 15 10 12
```



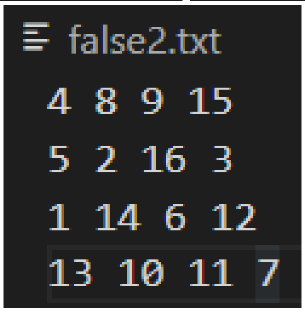
```
true2.txt
1 6 2 4
5 16 3 8
9 7 15 11
13 14 10 12
```



```
true3.txt
3 1 2 4
16 5 7 8
10 6 11 12
9 13 14 15
```

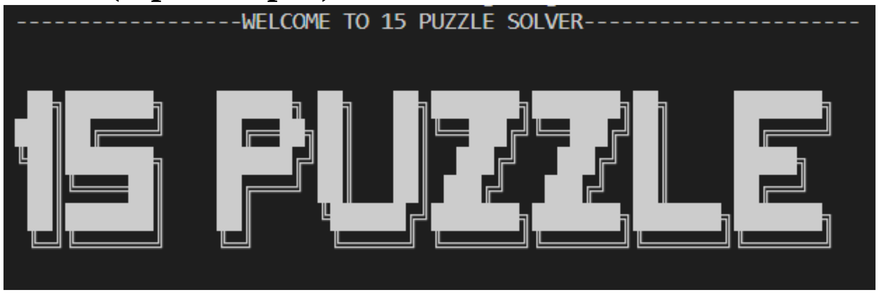


```
false1.txt
1 3 4 15
2 16 5 12
7 6 11 14
8 9 10 13
```



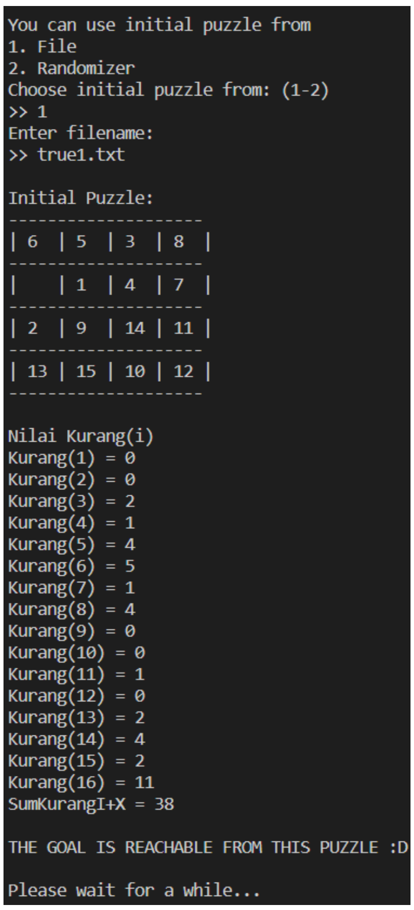
```
false2.txt
4 8 9 15
5 2 16 3
1 14 6 12
13 10 11 7
```

D. Hasil Eksekusi (input/output)

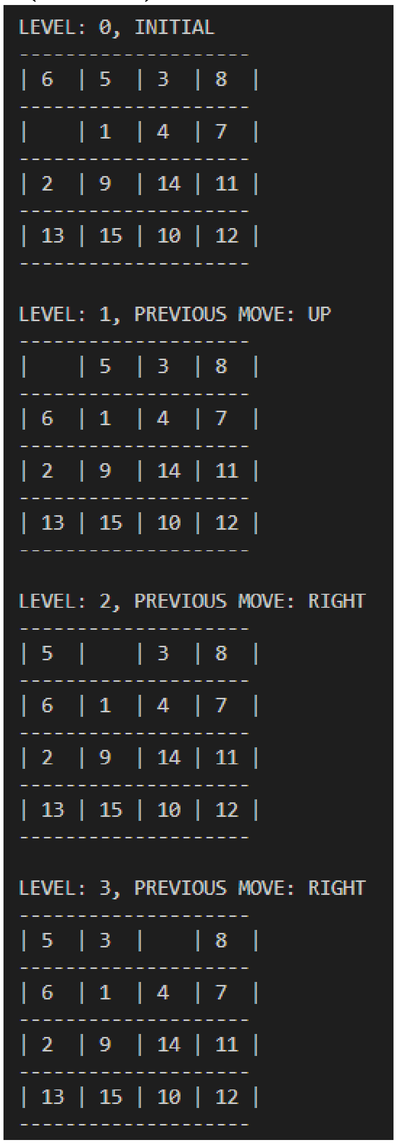


```
-----------------WELCOME TO 15 PUZZLE SOLVER---------------------
15 PUZZLE
```

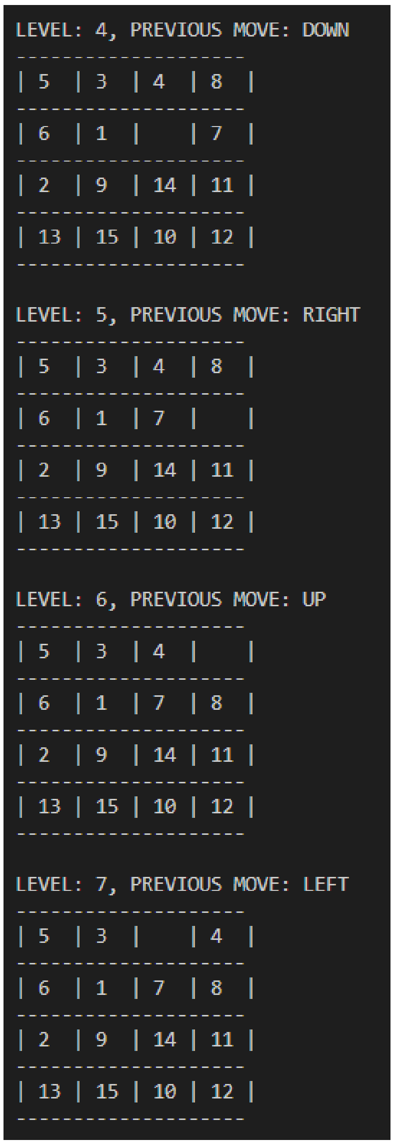
1. *Test case 1, reachable* (true1.txt)



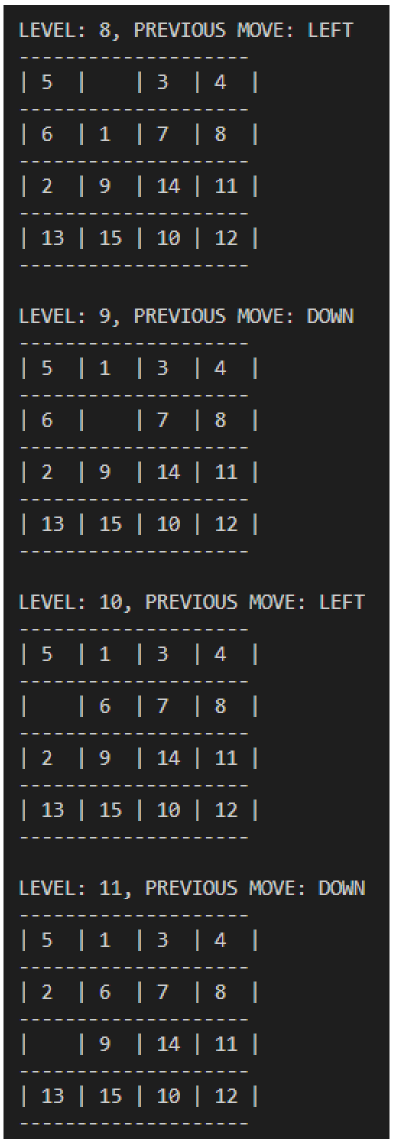
```
You can use initial puzzle from
1. File
2. Randomizer
Choose initial puzzle from: (1-2)
>> 1
Enter filename:
>> true1.txt

Initial Puzzle:
-------------------
| 6  | 5  | 3  | 8  |
-------------------
|    | 1  | 4  | 7  |
-------------------
| 2  | 9  | 14 | 11 |
-------------------
| 13 | 15 | 10 | 12 |
-------------------

Nilai Kurang(i)
Kurang(1) = 0
Kurang(2) = 0
Kurang(3) = 2
Kurang(4) = 1
Kurang(5) = 4
Kurang(6) = 5
Kurang(7) = 1
Kurang(8) = 4
Kurang(9) = 0
Kurang(10) = 0
Kurang(11) = 1
Kurang(12) = 0
Kurang(13) = 2
Kurang(14) = 4
Kurang(15) = 2
Kurang(16) = 11
SumKurangI+X = 38

THE GOAL IS REACHABLE FROM THIS PUZZLE :D

Please wait for a while...
```



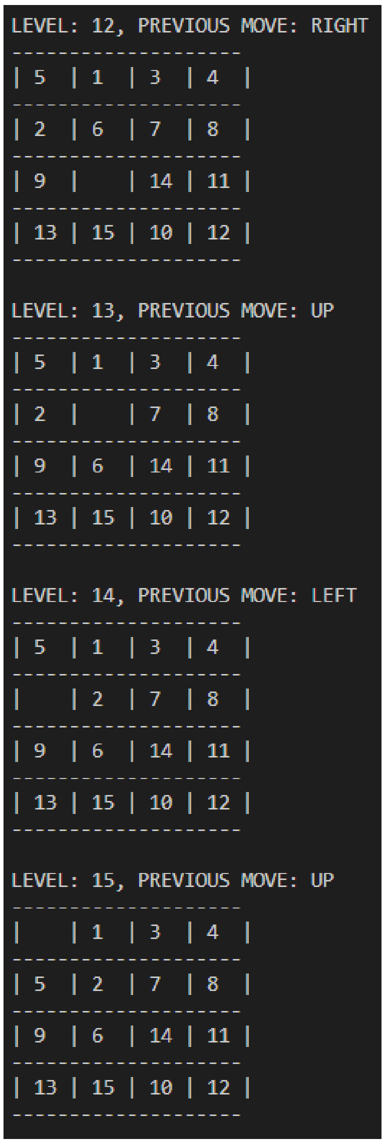
```
LEVEL: 0, INITIAL
--------------------
| 6  | 5  | 3  | 8  |
--------------------
|    | 1  | 4  | 7  |
--------------------
| 2  | 9  | 14 | 11 |
--------------------
| 13 | 15 | 10 | 12 |
--------------------

LEVEL: 1, PREVIOUS MOVE: UP
--------------------
|    | 5  | 3  | 8  |
--------------------
| 6  | 1  | 4  | 7  |
--------------------
| 2  | 9  | 14 | 11 |
--------------------
| 13 | 15 | 10 | 12 |
--------------------

LEVEL: 2, PREVIOUS MOVE: RIGHT
--------------------
| 5  |    | 3  | 8  |
--------------------
| 6  | 1  | 4  | 7  |
--------------------
| 2  | 9  | 14 | 11 |
--------------------
| 13 | 15 | 10 | 12 |
--------------------

LEVEL: 3, PREVIOUS MOVE: RIGHT
--------------------
| 5  | 3  |    | 8  |
--------------------
| 6  | 1  | 4  | 7  |
--------------------
| 2  | 9  | 14 | 11 |
--------------------
| 13 | 15 | 10 | 12 |
--------------------
```



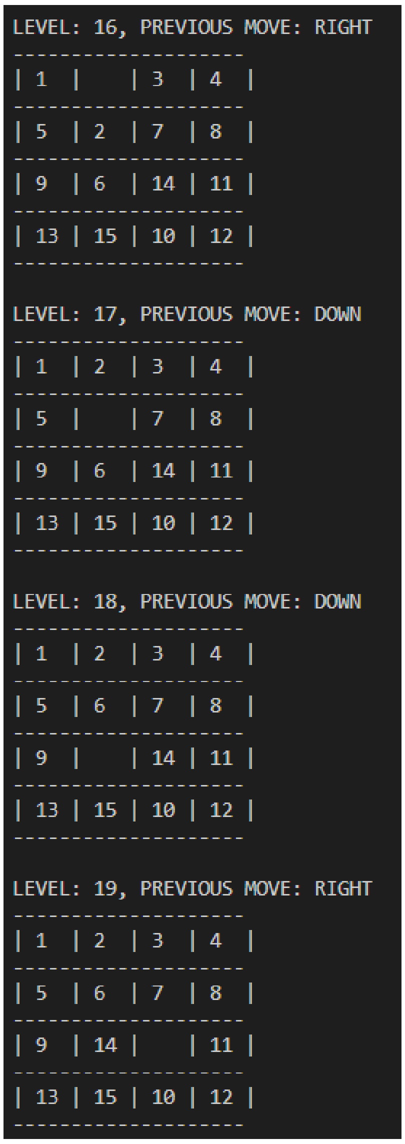
```
LEVEL: 4, PREVIOUS MOVE: DOWN
--------------------
| 5  | 3  | 4  | 8  |
--------------------
| 6  | 1  |    | 7  |
--------------------
| 2  | 9  | 14 | 11 |
--------------------
| 13 | 15 | 10 | 12 |
--------------------

LEVEL: 5, PREVIOUS MOVE: RIGHT
--------------------
| 5  | 3  | 4  | 8  |
--------------------
| 6  | 1  | 7  |    |
--------------------
| 2  | 9  | 14 | 11 |
--------------------
| 13 | 15 | 10 | 12 |
--------------------

LEVEL: 6, PREVIOUS MOVE: UP
--------------------
| 5  | 3  | 4  |    |
--------------------
| 6  | 1  | 7  | 8  |
--------------------
| 2  | 9  | 14 | 11 |
--------------------
| 13 | 15 | 10 | 12 |
--------------------

LEVEL: 7, PREVIOUS MOVE: LEFT
--------------------
| 5  | 3  |    | 4  |
--------------------
| 6  | 1  | 7  | 8  |
--------------------
| 2  | 9  | 14 | 11 |
--------------------
| 13 | 15 | 10 | 12 |
--------------------
```



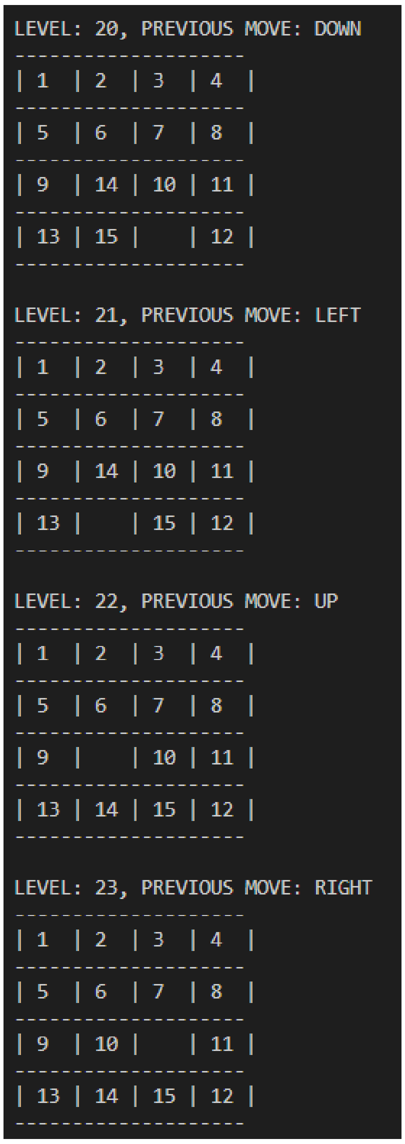
```
LEVEL: 8, PREVIOUS MOVE: LEFT
---------------------
| 5  |    | 3  | 4  |
---------------------
| 6  | 1  | 7  | 8  |
---------------------
| 2  | 9  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 9, PREVIOUS MOVE: DOWN
---------------------
| 5  | 1  | 3  | 4  |
---------------------
| 6  |    | 7  | 8  |
---------------------
| 2  | 9  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 10, PREVIOUS MOVE: LEFT
---------------------
| 5  | 1  | 3  | 4  |
---------------------
|    | 6  | 7  | 8  |
---------------------
| 2  | 9  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 11, PREVIOUS MOVE: DOWN
---------------------
| 5  | 1  | 3  | 4  |
---------------------
| 2  | 6  | 7  | 8  |
---------------------
|    | 9  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------
```



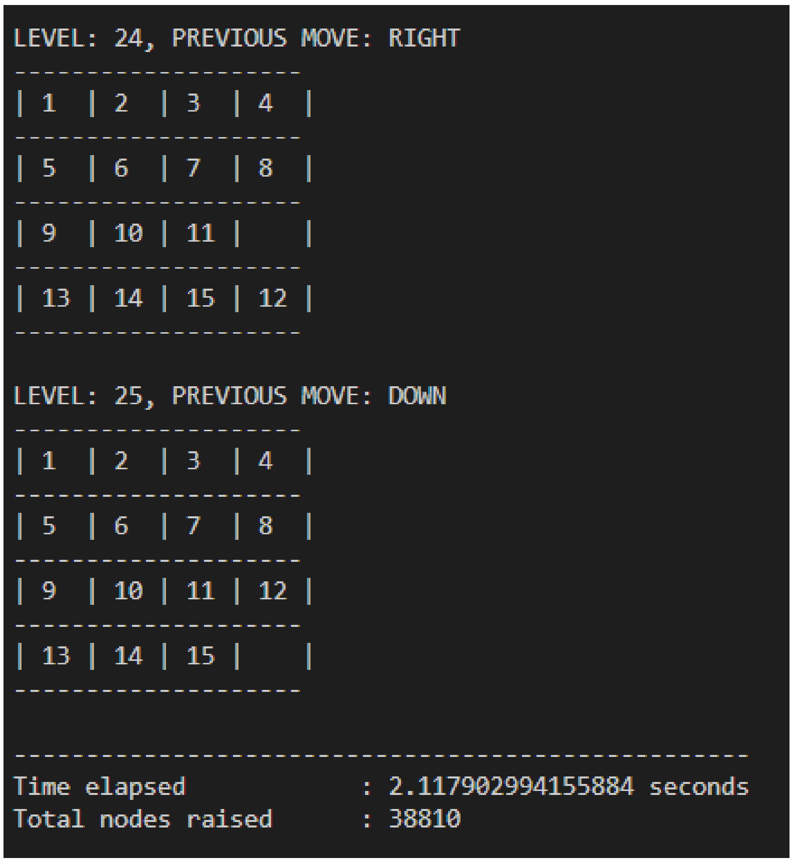
```
LEVEL: 12, PREVIOUS MOVE: RIGHT
---------------------
| 5  | 1  | 3  | 4  |
---------------------
| 2  | 6  | 7  | 8  |
---------------------
| 9  |    | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 13, PREVIOUS MOVE: UP
---------------------
| 5  | 1  | 3  | 4  |
---------------------
| 2  |    | 7  | 8  |
---------------------
| 9  | 6  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 14, PREVIOUS MOVE: LEFT
---------------------
| 5  | 1  | 3  | 4  |
---------------------
|    | 2  | 7  | 8  |
---------------------
| 9  | 6  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 15, PREVIOUS MOVE: UP
---------------------
|    | 1  | 3  | 4  |
---------------------
| 5  | 2  | 7  | 8  |
---------------------
| 9  | 6  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------
```



```
LEVEL: 16, PREVIOUS MOVE: RIGHT
---------------------
| 1  |    | 3  | 4  |
---------------------
| 5  | 2  | 7  | 8  |
---------------------
| 9  | 6  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 17, PREVIOUS MOVE: DOWN
---------------------
| 1  | 2  | 3  | 4  |
---------------------
| 5  |    | 7  | 8  |
---------------------
| 9  | 6  | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 18, PREVIOUS MOVE: DOWN
---------------------
| 1  | 2  | 3  | 4  |
---------------------
| 5  | 6  | 7  | 8  |
---------------------
| 9  |    | 14 | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------

LEVEL: 19, PREVIOUS MOVE: RIGHT
---------------------
| 1  | 2  | 3  | 4  |
---------------------
| 5  | 6  | 7  | 8  |
---------------------
| 9  | 14 |    | 11 |
---------------------
| 13 | 15 | 10 | 12 |
---------------------
```

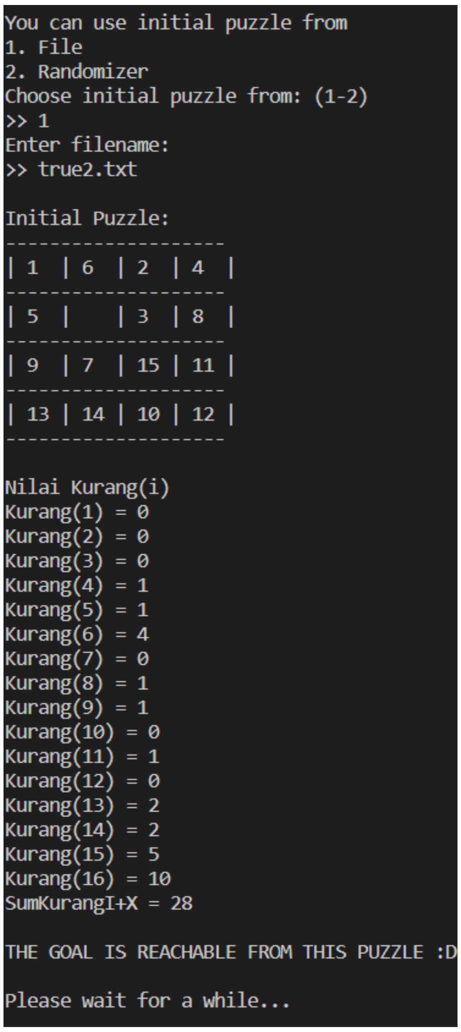


```
LEVEL: 20, PREVIOUS MOVE: DOWN
--------------------
| 1  | 2  | 3  | 4  |
--------------------
| 5  | 6  | 7  | 8  |
--------------------
| 9  | 14 | 10 | 11 |
--------------------
| 13 | 15 |    | 12 |
--------------------

LEVEL: 21, PREVIOUS MOVE: LEFT
--------------------
| 1  | 2  | 3  | 4  |
--------------------
| 5  | 6  | 7  | 8  |
--------------------
| 9  | 14 | 10 | 11 |
--------------------
| 13 |    | 15 | 12 |
--------------------

LEVEL: 22, PREVIOUS MOVE: UP
--------------------
| 1  | 2  | 3  | 4  |
--------------------
| 5  | 6  | 7  | 8  |
--------------------
| 9  |    | 10 | 11 |
--------------------
| 13 | 14 | 15 | 12 |
--------------------

LEVEL: 23, PREVIOUS MOVE: RIGHT
--------------------
| 1  | 2  | 3  | 4  |
--------------------
| 5  | 6  | 7  | 8  |
--------------------
| 9  | 10 |    | 11 |
--------------------
| 13 | 14 | 15 | 12 |
--------------------
```

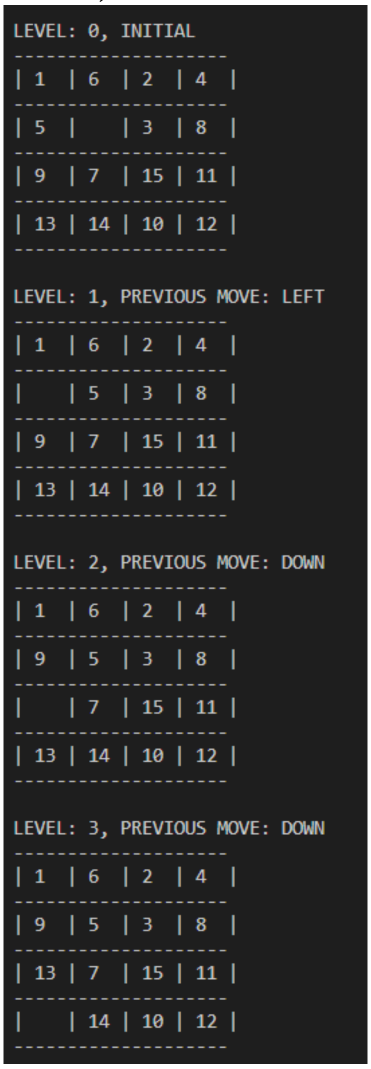


```
LEVEL: 24, PREVIOUS MOVE: RIGHT
--------------------
| 1  | 2  | 3  | 4  |
--------------------
| 5  | 6  | 7  | 8  |
--------------------
| 9  | 10 | 11 |    |
--------------------
| 13 | 14 | 15 | 12 |
--------------------

LEVEL: 25, PREVIOUS MOVE: DOWN
--------------------
| 1  | 2  | 3  | 4  |
--------------------
| 5  | 6  | 7  | 8  |
--------------------
| 9  | 10 | 11 | 12 |
--------------------
| 13 | 14 | 15 |    |
--------------------


---------------------------------------------------
Time elapsed           : 2.117902994155884 seconds
Total nodes raised     : 38810
```

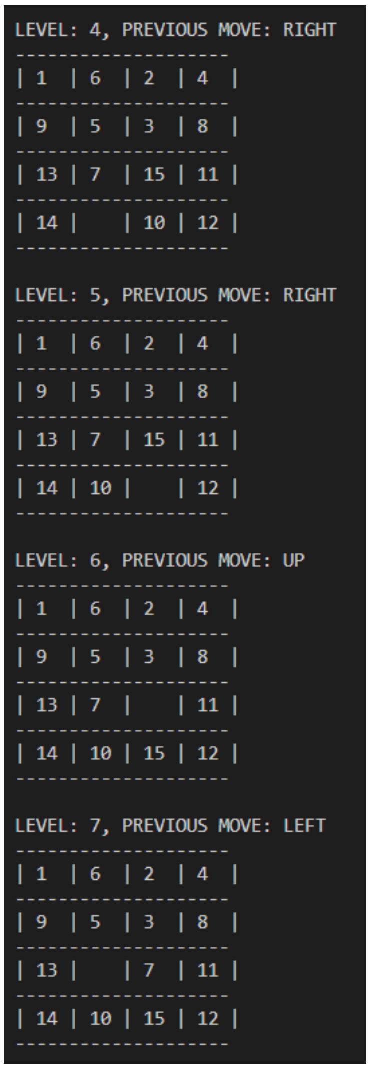
2. *Test case 2, reachable* (true2.txt)



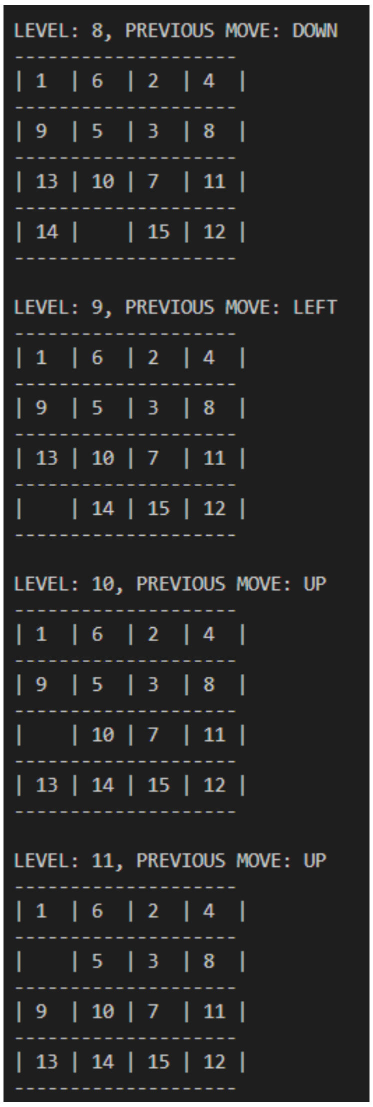
```
You can use initial puzzle from
1. File
2. Randomizer
Choose initial puzzle from: (1-2)
>> 1
Enter filename:
>> true2.txt

Initial Puzzle:
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 5  |    | 3  | 8  |
--------------------
| 9  | 7  | 15 | 11 |
--------------------
| 13 | 14 | 10 | 12 |
--------------------

Nilai Kurang(i)
Kurang(1) = 0
Kurang(2) = 0
Kurang(3) = 0
Kurang(4) = 1
Kurang(5) = 1
Kurang(6) = 4
Kurang(7) = 0
Kurang(8) = 1
Kurang(9) = 1
Kurang(10) = 0
Kurang(11) = 1
Kurang(12) = 0
Kurang(13) = 2
Kurang(14) = 2
Kurang(15) = 5
Kurang(16) = 10
SumKurangI+X = 28

THE GOAL IS REACHABLE FROM THIS PUZZLE :D

Please wait for a while...
```



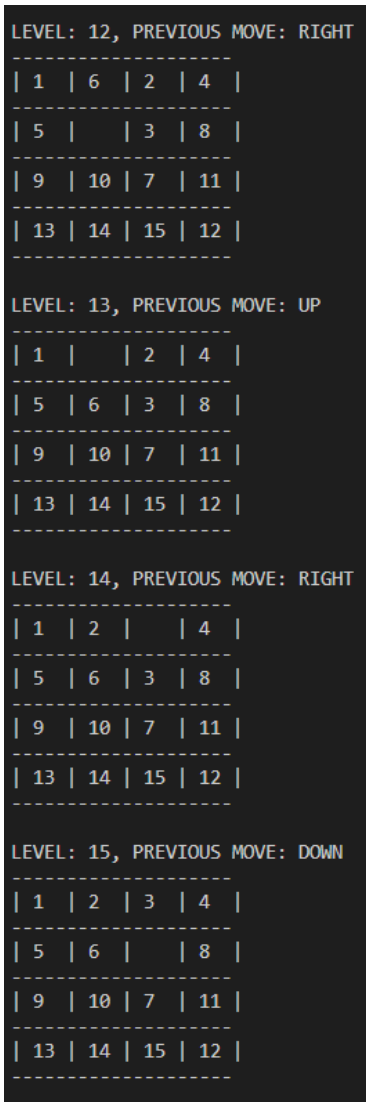
```
LEVEL: 0, INITIAL
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 5  |    | 3  | 8  |
--------------------
| 9  | 7  | 15 | 11 |
--------------------
| 13 | 14 | 10 | 12 |
--------------------

LEVEL: 1, PREVIOUS MOVE: LEFT
--------------------
| 1  | 6  | 2  | 4  |
--------------------
|    | 5  | 3  | 8  |
--------------------
| 9  | 7  | 15 | 11 |
--------------------
| 13 | 14 | 10 | 12 |
--------------------

LEVEL: 2, PREVIOUS MOVE: DOWN
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 9  | 5  | 3  | 8  |
--------------------
|    | 7  | 15 | 11 |
--------------------
| 13 | 14 | 10 | 12 |
--------------------

LEVEL: 3, PREVIOUS MOVE: DOWN
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 9  | 5  | 3  | 8  |
--------------------
| 13 | 7  | 15 | 11 |
--------------------
|    | 14 | 10 | 12 |
--------------------
```



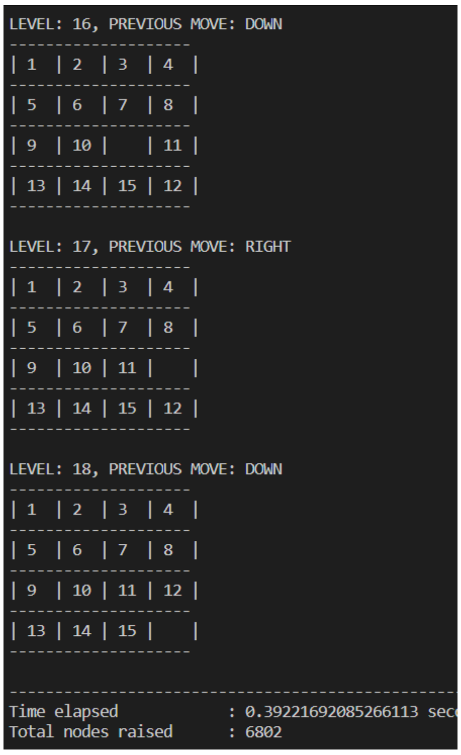
```
LEVEL: 4, PREVIOUS MOVE: RIGHT
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 9  | 5  | 3  | 8  |
--------------------
| 13 | 7  | 15 | 11 |
--------------------
| 14 |    | 10 | 12 |
--------------------

LEVEL: 5, PREVIOUS MOVE: RIGHT
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 9  | 5  | 3  | 8  |
--------------------
| 13 | 7  | 15 | 11 |
--------------------
| 14 | 10 |    | 12 |
--------------------

LEVEL: 6, PREVIOUS MOVE: UP
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 9  | 5  | 3  | 8  |
--------------------
| 13 | 7  |    | 11 |
--------------------
| 14 | 10 | 15 | 12 |
--------------------

LEVEL: 7, PREVIOUS MOVE: LEFT
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 9  | 5  | 3  | 8  |
--------------------
| 13 |    | 7  | 11 |
--------------------
| 14 | 10 | 15 | 12 |
--------------------
```



```
LEVEL: 8, PREVIOUS MOVE: DOWN
---------------------
| 1  | 6  | 2  | 4  |
---------------------
| 9  | 5  | 3  | 8  |
---------------------
| 13 | 10 | 7  | 11 |
---------------------
| 14 |    | 15 | 12 |
---------------------

LEVEL: 9, PREVIOUS MOVE: LEFT
---------------------
| 1  | 6  | 2  | 4  |
---------------------
| 9  | 5  | 3  | 8  |
---------------------
| 13 | 10 | 7  | 11 |
---------------------
|    | 14 | 15 | 12 |
---------------------

LEVEL: 10, PREVIOUS MOVE: UP
---------------------
| 1  | 6  | 2  | 4  |
---------------------
| 9  | 5  | 3  | 8  |
---------------------
|    | 10 | 7  | 11 |
---------------------
| 13 | 14 | 15 | 12 |
---------------------

LEVEL: 11, PREVIOUS MOVE: UP
---------------------
| 1  | 6  | 2  | 4  |
---------------------
|    | 5  | 3  | 8  |
---------------------
| 9  | 10 | 7  | 11 |
---------------------
| 13 | 14 | 15 | 12 |
---------------------
```

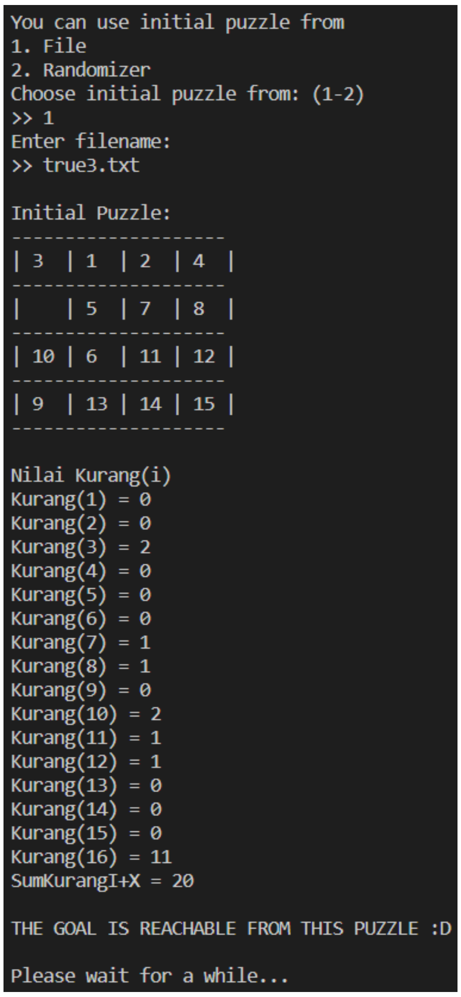


```
LEVEL: 12, PREVIOUS MOVE: RIGHT
---------------------
| 1  | 6  | 2  | 4  |
---------------------
| 5  |    | 3  | 8  |
---------------------
| 9  | 10 | 7  | 11 |
---------------------
| 13 | 14 | 15 | 12 |
---------------------

LEVEL: 13, PREVIOUS MOVE: UP
---------------------
| 1  |    | 2  | 4  |
---------------------
| 5  | 6  | 3  | 8  |
---------------------
| 9  | 10 | 7  | 11 |
---------------------
| 13 | 14 | 15 | 12 |
---------------------

LEVEL: 14, PREVIOUS MOVE: RIGHT
---------------------
| 1  | 2  |    | 4  |
---------------------
| 5  | 6  | 3  | 8  |
---------------------
| 9  | 10 | 7  | 11 |
---------------------
| 13 | 14 | 15 | 12 |
---------------------

LEVEL: 15, PREVIOUS MOVE: DOWN
---------------------
| 1  | 2  | 3  | 4  |
---------------------
| 5  | 6  |    | 8  |
---------------------
| 9  | 10 | 7  | 11 |
---------------------
| 13 | 14 | 15 | 12 |
---------------------
```

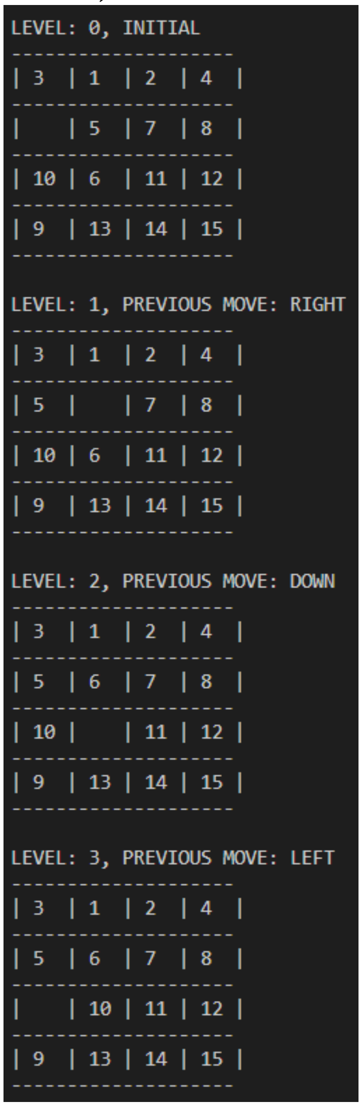


```
LEVEL: 16, PREVIOUS MOVE: DOWN
---------------------
| 1  | 2  | 3  | 4  |
---------------------
| 5  | 6  | 7  | 8  |
---------------------
| 9  | 10 |    | 11 |
---------------------
| 13 | 14 | 15 | 12 |
---------------------

LEVEL: 17, PREVIOUS MOVE: RIGHT
---------------------
| 1  | 2  | 3  | 4  |
---------------------
| 5  | 6  | 7  | 8  |
---------------------
| 9  | 10 | 11 |    |
---------------------
| 13 | 14 | 15 | 12 |
---------------------

LEVEL: 18, PREVIOUS MOVE: DOWN
---------------------
| 1  | 2  | 3  | 4  |
---------------------
| 5  | 6  | 7  | 8  |
---------------------
| 9  | 10 | 11 | 12 |
---------------------
| 13 | 14 | 15 |    |
---------------------

---------------------------------------------------
Time elapsed          : 0.39221692085266113 sec
Total nodes raised    : 6802
```

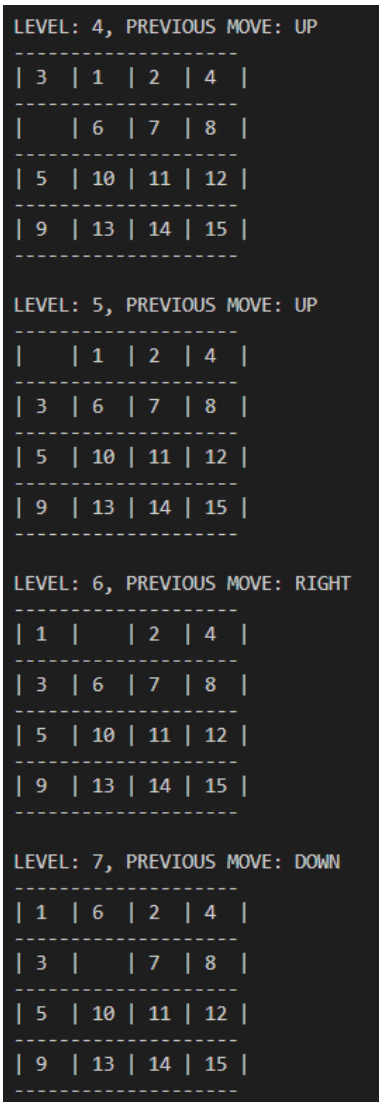
3. *Test case 3, reachable* (true3.txt)



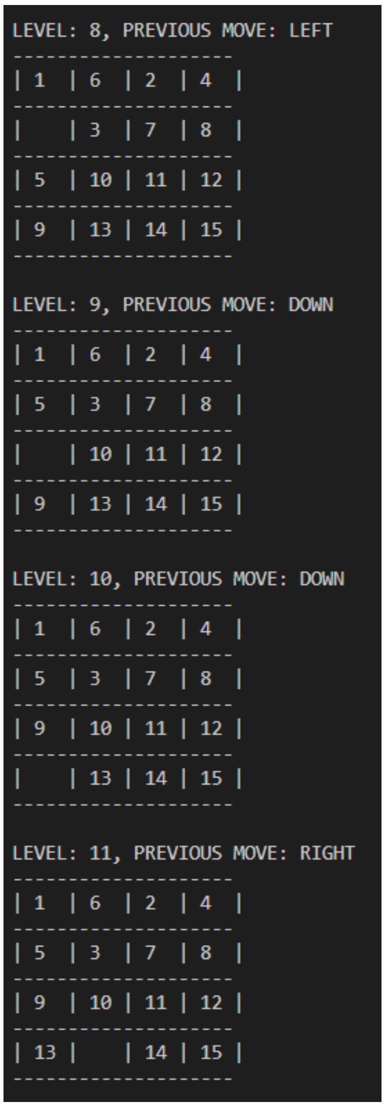
```
You can use initial puzzle from
1. File
2. Randomizer
Choose initial puzzle from: (1-2)
>> 1
Enter filename:
>> true3.txt

Initial Puzzle:
--------------------
| 3  | 1  | 2  | 4  |
--------------------
|    | 5  | 7  | 8  |
--------------------
| 10 | 6  | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------

Nilai Kurang(i)
Kurang(1) = 0
Kurang(2) = 0
Kurang(3) = 2
Kurang(4) = 0
Kurang(5) = 0
Kurang(6) = 0
Kurang(7) = 1
Kurang(8) = 1
Kurang(9) = 0
Kurang(10) = 2
Kurang(11) = 1
Kurang(12) = 1
Kurang(13) = 0
Kurang(14) = 0
Kurang(15) = 0
Kurang(16) = 11
SumKurangI+X = 20

THE GOAL IS REACHABLE FROM THIS PUZZLE :D

Please wait for a while...
```



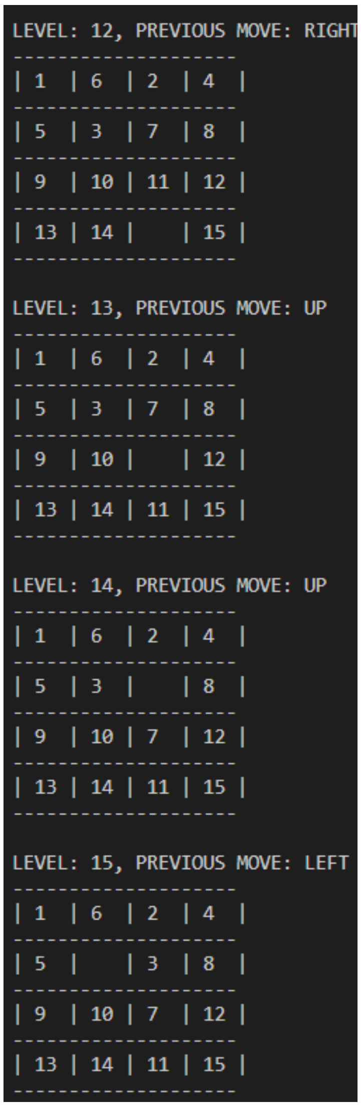
```
LEVEL: 0, INITIAL
--------------------
| 3  | 1  | 2  | 4  |
--------------------
|    | 5  | 7  | 8  |
--------------------
| 10 | 6  | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------

LEVEL: 1, PREVIOUS MOVE: RIGHT
--------------------
| 3  | 1  | 2  | 4  |
--------------------
| 5  |    | 7  | 8  |
--------------------
| 10 | 6  | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------

LEVEL: 2, PREVIOUS MOVE: DOWN
--------------------
| 3  | 1  | 2  | 4  |
--------------------
| 5  | 6  | 7  | 8  |
--------------------
| 10 |    | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------

LEVEL: 3, PREVIOUS MOVE: LEFT
--------------------
| 3  | 1  | 2  | 4  |
--------------------
| 5  | 6  | 7  | 8  |
--------------------
|    | 10 | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------
```



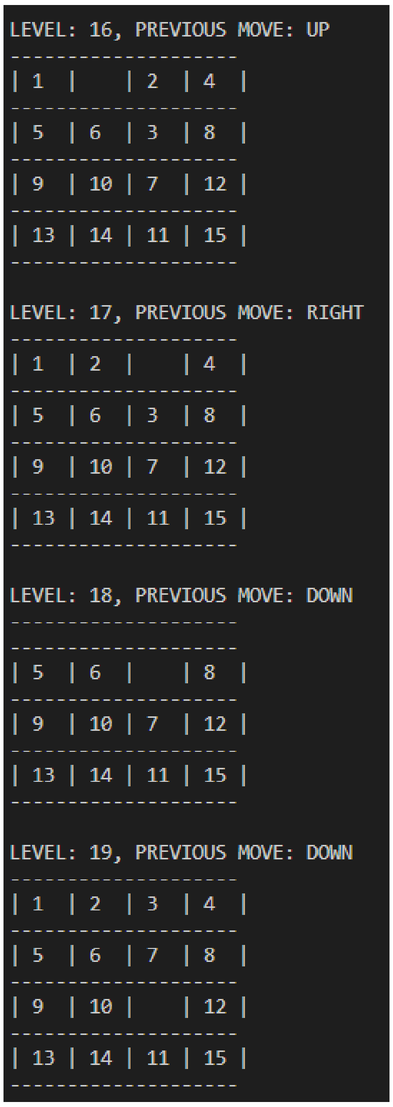
```
LEVEL: 4, PREVIOUS MOVE: UP
--------------------
| 3  | 1  | 2  | 4  |
--------------------
|    | 6  | 7  | 8  |
--------------------
| 5  | 10 | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------

LEVEL: 5, PREVIOUS MOVE: UP
--------------------
|    | 1  | 2  | 4  |
--------------------
| 3  | 6  | 7  | 8  |
--------------------
| 5  | 10 | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------

LEVEL: 6, PREVIOUS MOVE: RIGHT
--------------------
| 1  |    | 2  | 4  |
--------------------
| 3  | 6  | 7  | 8  |
--------------------
| 5  | 10 | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------

LEVEL: 7, PREVIOUS MOVE: DOWN
--------------------
| 1  | 6  | 2  | 4  |
--------------------
| 3  |    | 7  | 8  |
--------------------
| 5  | 10 | 11 | 12 |
--------------------
| 9  | 13 | 14 | 15 |
--------------------
```



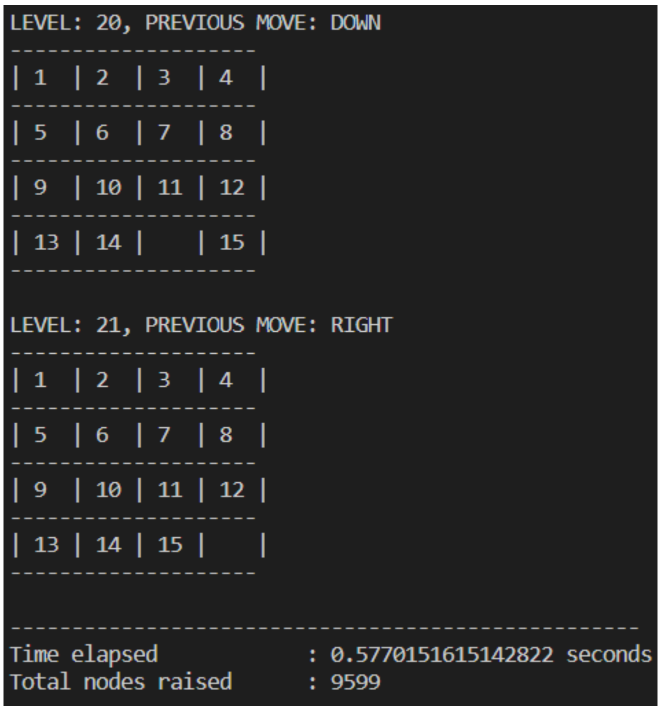
```
LEVEL: 8, PREVIOUS MOVE: LEFT
-------------------
| 1  | 6  | 2  | 4  |
-------------------
|    | 3  | 7  | 8  |
-------------------
| 5  | 10 | 11 | 12 |
-------------------
| 9  | 13 | 14 | 15 |
-------------------

LEVEL: 9, PREVIOUS MOVE: DOWN
-------------------
| 1  | 6  | 2  | 4  |
-------------------
| 5  | 3  | 7  | 8  |
-------------------
|    | 10 | 11 | 12 |
-------------------
| 9  | 13 | 14 | 15 |
-------------------

LEVEL: 10, PREVIOUS MOVE: DOWN
-------------------
| 1  | 6  | 2  | 4  |
-------------------
| 5  | 3  | 7  | 8  |
-------------------
| 9  | 10 | 11 | 12 |
-------------------
|    | 13 | 14 | 15 |
-------------------

LEVEL: 11, PREVIOUS MOVE: RIGHT
-------------------
| 1  | 6  | 2  | 4  |
-------------------
| 5  | 3  | 7  | 8  |
-------------------
| 9  | 10 | 11 | 12 |
-------------------
| 13 |    | 14 | 15 |
-------------------
```



```
LEVEL: 12, PREVIOUS MOVE: RIGHT
-------------------
| 1  | 6  | 2  | 4  |
-------------------
| 5  | 3  | 7  | 8  |
-------------------
| 9  | 10 | 11 | 12 |
-------------------
| 13 | 14 |    | 15 |
-------------------

LEVEL: 13, PREVIOUS MOVE: UP
-------------------
| 1  | 6  | 2  | 4  |
-------------------
| 5  | 3  | 7  | 8  |
-------------------
| 9  | 10 |    | 12 |
-------------------
| 13 | 14 | 11 | 15 |
-------------------

LEVEL: 14, PREVIOUS MOVE: UP
-------------------
| 1  | 6  | 2  | 4  |
-------------------
| 5  | 3  |    | 8  |
-------------------
| 9  | 10 | 7  | 12 |
-------------------
| 13 | 14 | 11 | 15 |
-------------------

LEVEL: 15, PREVIOUS MOVE: LEFT
-------------------
| 1  | 6  | 2  | 4  |
-------------------
| 5  |    | 3  | 8  |
-------------------
| 9  | 10 | 7  | 12 |
-------------------
| 13 | 14 | 11 | 15 |
-------------------
```

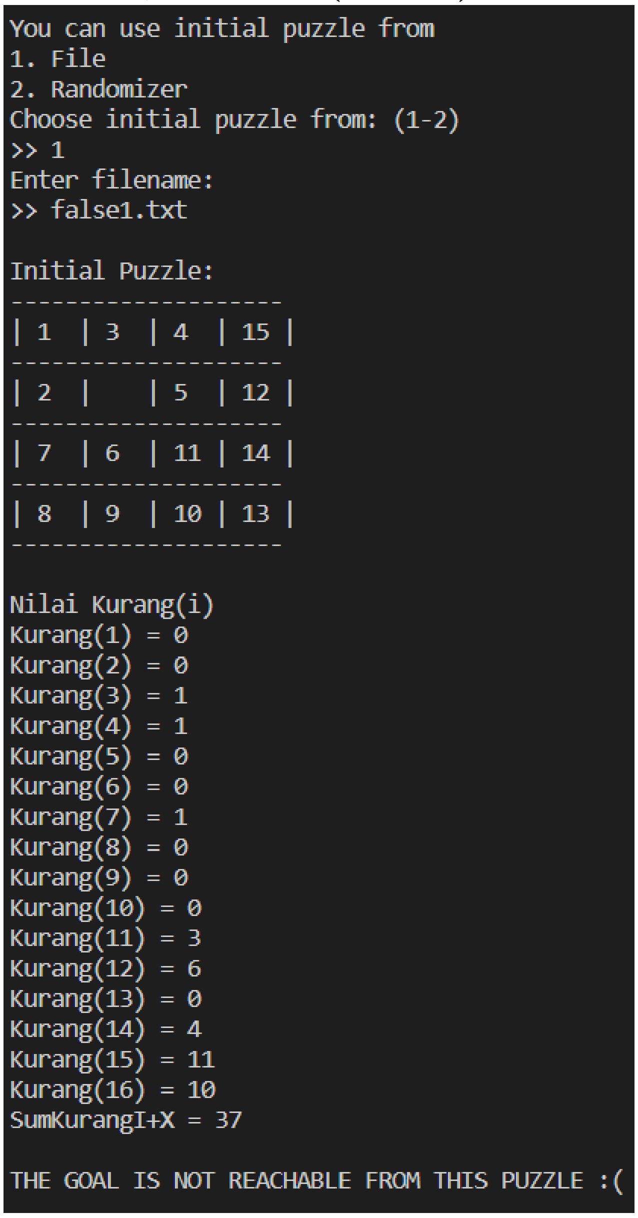


```
LEVEL: 16, PREVIOUS MOVE: UP
-------------------
| 1  |    | 2  | 4  |
-------------------
| 5  | 6  | 3  | 8  |
-------------------
| 9  | 10 | 7  | 12 |
-------------------
| 13 | 14 | 11 | 15 |
-------------------

LEVEL: 17, PREVIOUS MOVE: RIGHT
-------------------
| 1  | 2  |    | 4  |
-------------------
| 5  | 6  | 3  | 8  |
-------------------
| 9  | 10 | 7  | 12 |
-------------------
| 13 | 14 | 11 | 15 |
-------------------

LEVEL: 18, PREVIOUS MOVE: DOWN
-------------------
-------------------
| 5  | 6  |    | 8  |
-------------------
| 9  | 10 | 7  | 12 |
-------------------
| 13 | 14 | 11 | 15 |
-------------------

LEVEL: 19, PREVIOUS MOVE: DOWN
-------------------
| 1  | 2  | 3  | 4  |
-------------------
| 5  | 6  | 7  | 8  |
-------------------
| 9  | 10 |    | 12 |
-------------------
| 13 | 14 | 11 | 15 |
-------------------
```



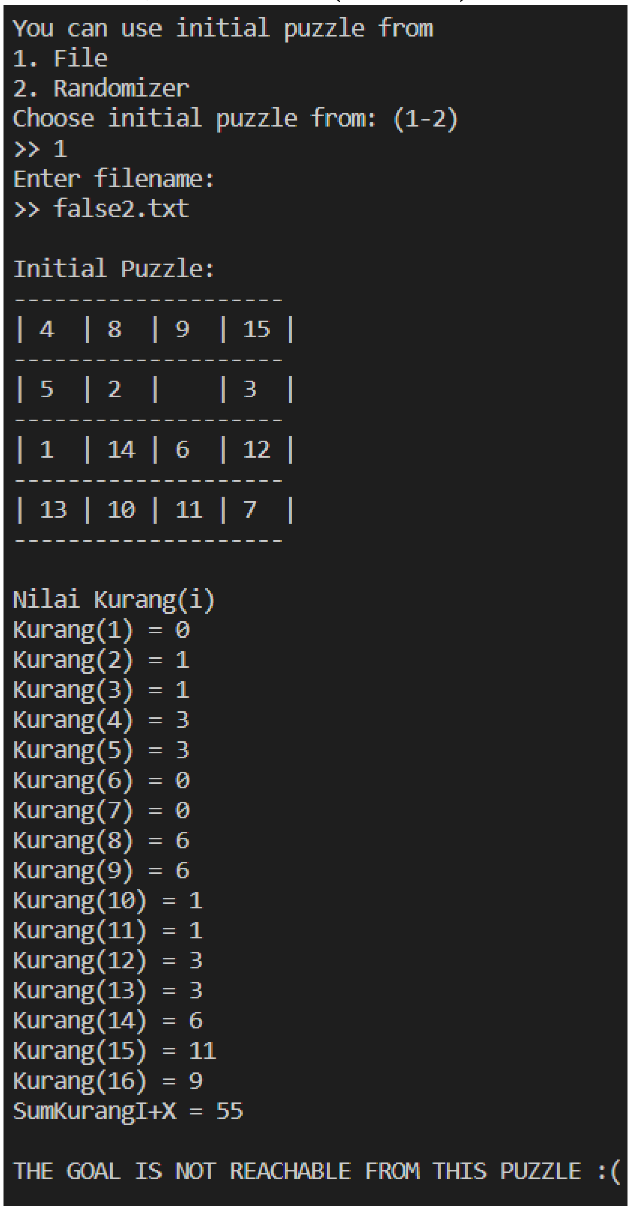
```
LEVEL: 20, PREVIOUS MOVE: DOWN
-------------------
| 1  | 2  | 3  | 4  |
-------------------
| 5  | 6  | 7  | 8  |
-------------------
| 9  | 10 | 11 | 12 |
-------------------
| 13 | 14 |    | 15 |
-------------------

LEVEL: 21, PREVIOUS MOVE: RIGHT
-------------------
| 1  | 2  | 3  | 4  |
-------------------
| 5  | 6  | 7  | 8  |
-------------------
| 9  | 10 | 11 | 12 |
-------------------
| 13 | 14 | 15 |    |
-------------------

--------------------------------------------------
Time elapsed            : 0.5770151615142822 seconds
Total nodes raised      : 9599
```

4. *Test case 4, unreachable* (false1.txt)



```
You can use initial puzzle from
1. File
2. Randomizer
Choose initial puzzle from: (1-2)
>> 1
Enter filename:
>> false1.txt

Initial Puzzle:
--------------------
| 1  | 3  | 4  | 15 |
--------------------
| 2  |    | 5  | 12 |
--------------------
| 7  | 6  | 11 | 14 |
--------------------
| 8  | 9  | 10 | 13 |
--------------------

Nilai Kurang(i)
Kurang(1) = 0
Kurang(2) = 0
Kurang(3) = 1
Kurang(4) = 1
Kurang(5) = 0
Kurang(6) = 0
Kurang(7) = 1
Kurang(8) = 0
Kurang(9) = 0
Kurang(10) = 0
Kurang(11) = 3
Kurang(12) = 6
Kurang(13) = 0
Kurang(14) = 4
Kurang(15) = 11
Kurang(16) = 10
SumKurangI+X = 37

THE GOAL IS NOT REACHABLE FROM THIS PUZZLE :(
```

5. *Test case 5, unreachable* (false2.txt)



```
You can use initial puzzle from
1. File
2. Randomizer
Choose initial puzzle from: (1-2)
>> 1
Enter filename:
>> false2.txt

Initial Puzzle:
--------------------
| 4  | 8  | 9  | 15 |
--------------------
| 5  | 2  |    | 3  |
--------------------
| 1  | 14 | 6  | 12 |
--------------------
| 13 | 10 | 11 | 7  |
--------------------

Nilai Kurang(i)
Kurang(1) = 0
Kurang(2) = 1
Kurang(3) = 1
Kurang(4) = 3
Kurang(5) = 3
Kurang(6) = 0
Kurang(7) = 0
Kurang(8) = 6
Kurang(9) = 6
Kurang(10) = 1
Kurang(11) = 1
Kurang(12) = 3
Kurang(13) = 3
Kurang(14) = 6
Kurang(15) = 11
Kurang(16) = 9
SumKurangI+X = 55

THE GOAL IS NOT REACHABLE FROM THIS PUZZLE :(
```

E. *Repository* Program (Github)

Untuk menjalankan program, *source code* bisa didapatkan di

https://github.com/Putriliza/Tucil3_13520066.git

F. Rangkuman Keberjalanan Program

Poin	Ya	Tidak
1. Program berhasil dikompilasi	✓	-
2. Program berhasil running	✓	-
3. Program dapat menerima input dan menuliskan output.	✓	-
4. Luaran sudah benar untuk semua data uji	✓	-
5. Bonus dibuat	-	✓