**Operating Systems – Lab**

**Mini Project**

**BCS – 5D**

**21F – 9137**

**21F – 9116**

|  |
| --- |
| **Mini Project** |
| **Bash-Script (Code):**  #!/bin/bash  declare -a initState  declare -a goalState  size=5  moves=""  # Initial state (function)  init() {  numbers=( $(seq 1 $((size\*size - 1))) 0 )  for i in $(seq 1 100); do  j=$((RANDOM % (size\*size)))  k=$((RANDOM % (size\*size)))  temp=${numbers[j]}  numbers[j]=${numbers[k]}  numbers[k]=$temp  done  initState=("${numbers[@]}")  }  # Initialize the goal state (function)  goal() {  numbers=( $(seq 1 $((size\*size - 1))) 0 )  for i in $(seq 1 100); do  j=$((RANDOM % (size\*size)))  k=$((RANDOM % (size\*size)))  temp=${numbers[j]}  numbers[j]=${numbers[k]}  numbers[k]=$temp  done  goalState=("${numbers[@]}")  }  # Print the board state (function)  print\_board() {  local -n arr=$1  for ((i=0; i<size; i++)); do  for ((j=0; j<size; j++)); do  printf "%2d " ${arr[size\*i + j]}  done  echo ""  done  echo ""  }  # Check if the current state matches the goal state (function)  is\_goal() {  for ((i=0; i<${#initState[@]}; i++)); do  if [[ ${initState[i]} -ne ${goalState[i]} ]]; then  return 1  fi  done  return 0  }  # Get legal moves (function)  legal\_moves() {  moves=""  for ((i=0; i<size\*size; i++)); do  if [[ ${initState[i]} -eq 0 ]]; then  if ((i % size > 0)); then moves+="L"; fi  if ((i % size < size - 1)); then moves+="R"; fi  if ((i / size > 0)); then moves+="U"; fi  if ((i / size < size - 1)); then moves+="D"; fi  break  fi  done  }  # Make a move (function)  make\_move() {  for ((i=0; i<size\*size; i++)); do  if [[ ${initState[i]} -eq 0 ]]; then  case $1 in  L) j=$((i-1));;  R) j=$((i+1));;  U) j=$((i-size));;  D) j=$((i+size));;  esac  initState[i]=${initState[j]}  initState[j]=0  return  fi  done  }  # Now on, this all serves as main part  # Initialize the states  init  goal  # Game's while loop  while true; do  clear  echo "Current State:"  print\_board initState    echo "Goal State:"  print\_board goalState    is\_goal && { echo "Congratulations! You solved the puzzle!"; exit 0; }  legal\_moves  echo "Legal moves: $moves"  echo -n "Your move (L/R/U/D) or Q to quit: "  read -n 1 move  echo ""  if [[ $move == "Q" ]]; then  exit 0  elif [[ $moves == \*$move\* ]]; then  make\_move $move  else  echo "Invalid move!"  sleep 1  fi  done  # Game ends |
| **Outputs:** |

**Explanation:** When the game begins, two boards will be printed at random, one will be initial board and other will be goal board. The goal board will remain same and printed whilst the initial board will welcome moves from the player. The legal moves will keep on printing until the game ends. Then when the game ends, it will print the moves made as well.