Class Assignment

Course Title: Operating System

Course Code: CSE 309

Section: PC DA

Submitted to
Dr. Faiz Al Faisal
Assistant Professor
Dept. of CSE
Green University of Bangladesh

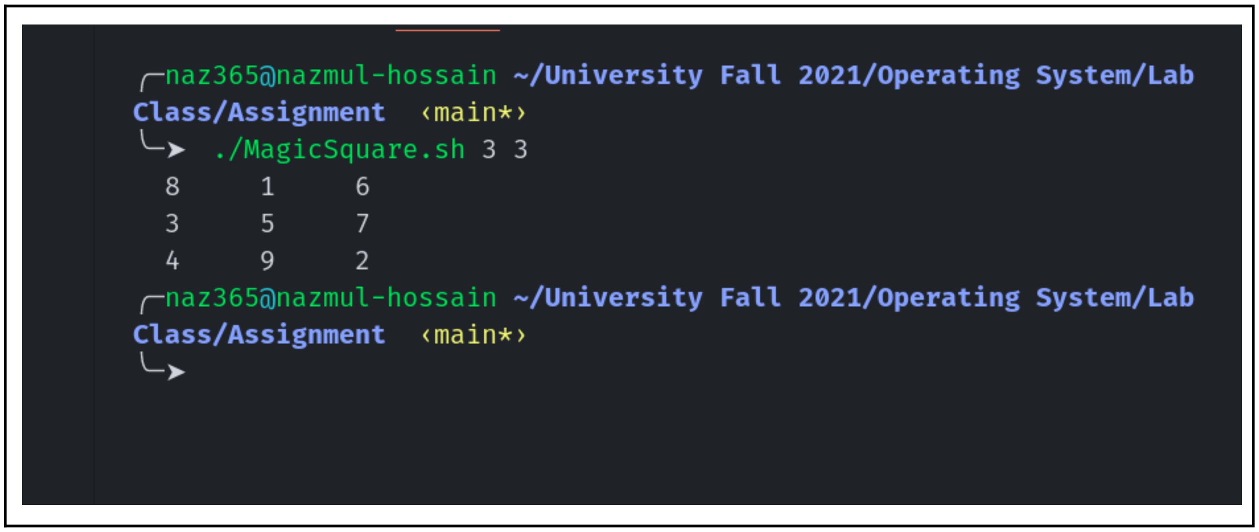
Submitted by:
Mohammad Nazmul Hossain
ID:193902031
Dept. of CSE

Code:

```bash
#!/bin/bash


# Definition: A "magic square" is a two-dimensional array
#              of integers in which all the rows, columns,
#              and *long* diagonals add up to the same number.
#              Being "square," the array has the same number
#              of rows and columns.
# An example of a magic square of order 3 is:
#   8   1   6
#   3   5   7
#   4   9   2
# All the rows, columns, and long diagonals add up to 15.


# Globals
EVEN=2
MAXSIZE=31   # 31 rows x 31 cols.
E_usage=90   # Invocation error.
dimension=
declare -i square

usage_message ()
{
 echo "Usage: $0 square-size"
 echo "   ... where \"square-size\" is an ODD integer"
 echo "       in the range 3 - 31."
  exit $E_usage
}
```

```bash
calculate ()       # Here's where the actual work gets done.
{
 local row col index dimadj j k cell_val=1
 dimension=$1

 let "dimadj = $dimension * 3"; let "dimadj /= 2"   # x 1.5, then
truncate.

 for ((j=0; j < dimension; j++))
 do
   for ((k=0; k < dimension; k++))
   do  # Calculate indices, then convert to 1-dim. array index.
       # Bash doesn't support multidimensional arrays. Pity.
     let "col = $k - $j + $dimadj"; let "col %= $dimension"
     let "row = $j * 2 - $k + $dimension"; let "row %= $dimension"
     let "index = $row*($dimension) + $col"
     square[$index]=cell_val; ((cell_val++))
   done
 done
}     # Plain math, no visualization required.


print_square ()               # Output square, one row at a time.
{
 local row col idx d1
 let "d1 = $dimension - 1"   # Adjust for zero-indexed array.
 for row in $(seq 0 $d1)
 do

   for col in $(seq 0 $d1)
   do
     let "idx = $row * $dimension + $col"
     printf "%3d " "${square[idx]}"; echo -n "  "
   done   # Displays up to 13-order neatly in 80-column term window.

   echo   # Newline after each row.
 done
}



#################################################
if [[ -z "$1" ]] || [[ "$1" -gt $MAXSIZE ]]
then
 usage_message
fi

let "test_even = $1 % $EVEN"
if [ $test_even -eq 0 ]
then           # Can't handle even-order squares.
 usage_message
fi

calculate $1
print_square   # echo "${square[@]}"   # DEBUG

exit $?
#################################################
```

Output:



```
╭─naz365@nazmul-hossain ~/University Fall 2021/Operating System/Lab
Class/Assignment  ‹main*›
╰─➤  ./MagicSquare.sh 3 3
  8      1      6
  3      5      7
  4      9      2
╭─naz365@nazmul-hossain ~/University Fall 2021/Operating System/Lab
Class/Assignment  ‹main*›
╰─➤
```