



**COURSE NAME:** ALGORITHM ANALYSIS

**COURSE INSTRUCTOR:** Dr. M. Amaç GÜVENSAN

**STUDENT NAME:** Ertuğrul ŞENTÜRK

**STUDENT NO:** 18011028

**STUDENT E-MAIL:** [mdesenturk@gmail.com](mailto:mdesenturk@gmail.com)

**SCHOOL YEAR:** 3

**GROUP NUMBER:** 1

**HOMEWORK NUMBER:** 1

**QUESTION NUMBER:** 2

**QUESTION:** Creating a matrix of  $n$  value given according to the rule of "Von Neumann's Neighborhood"

## ALGORITHM:

- The value of  $n$  received from the user.
- $2 * n + 3$  size square matrix was initially created with dynamic memory allocation using `calloc` since all elements should be 0.
- An array of the same size was created to store the number of black squares in each row.
- According to the algorithm, the starting and ending points of the black squares in each row were calculated.
- Since the black squares increase in the upper half and decrease in the lower half, the starting and ending values of the squares were calculated according to the relevant situation.
- Each row is filled in according to the calculated start and end values.
- Since 1 more of the difference between the start and end values will give us the number of black squares in that row, this value has been recorded in the array that holds the number of black squares in the row.
- Matrix and array printed to user.
- The total number of black squares was calculated and printed according to the relevant formula.

## PROGRAM OUTPUTS:

```
Please enter a n value: 0
Results:

Output Matrix:
| 0 0 0 |
| 0 1 0 |
| 0 0 0 |

Black Squares in Rows: [0, 1, 0]
Total number of black squares:: 1
```

for n = 0

```
Please enter a n value: 3
Results:

Output Matrix:
| 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 1 0 0 0 0 |
| 0 0 0 1 1 1 0 0 0 |
| 0 0 1 1 1 1 1 0 0 |
| 0 1 1 1 1 1 1 1 0 |
| 0 0 1 1 1 1 1 0 0 |
| 0 0 0 1 1 1 0 0 0 |
| 0 0 0 0 1 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 |

Black Squares in Rows: [0, 1, 3, 5, 7, 5, 3, 1, 0]
Total number of black squares:: 25
```

for n = 3

```
Please enter a n value: 7
Results:

Output Matrix:
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 |
| 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 |
| 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 |
| 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 |
| 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 |
| 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 |
| 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 |
| 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 |
| 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 |
| 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 |
| 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 |
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |

Black Squares in Rows: [0, 1, 3, 5, 7, 9, 11, 13, 15, 13, 11, 9, 7, 5, 3, 1, 0]
Total number of black squares:: 113
```

for n = 7