

**CSE 331-503**

**PROJECT 1 REPORT**

**BOMBERMAN**

*Muhammed Yasir Güneş*

*210104004079*

## 1. Input Reading

The program efficiently reads user input through one string system call.

The user is prompted to provide the row size, column size, seconds, and the initial grid configuration.

System calls 5 (read integer) and 8 (read string) are employed for obtaining numeric and string inputs, respectively.

These inputs are then stored in memory locations reserved for row, column, seconds, and the grid arrays.

```
Enter the row size:
5
Enter the column size:
5
Enter the seconds: 18
```

When reading the grid array. User should only insert “.” or “O”. User have to insert the grid array in one line. Like this:

```
Enter the grid:
..O..OO.O..O.O..O.O....O
```

If the grid the user want is like this:

```
Grid:
..O..
OO.O.
.O.O.
.O.O.
...O
```

## **2. Program Operation:**

### **2.1. Overview:**

The Bomberman program is designed to simulate the planting and exploding of bombs in a grid-based environment. It uses two grids (grid1 and grid2) to represent the current state of the grid and the next state after bomb placement and explosion.

### **2.2. Main Functions:**

#### **2.2.1. bomberMan Function:**

This function iterates through each cell of the grid and places a bomb ('O') if the cell is empty ('.'). It utilizes a loop to traverse the grid and conditionally sets cells to 'O' based on the existing contents.

This function is implemented on the grid2 which is the copy of grid1.

grid1 is to keep the initial state and compare it.

grid2 is to implement functions on it according to the grid1.

#### **2.2.2. explodeBombs Function:**

Responsible for simulating the explosion of bombs in the grid. It checks each cell in the input grid (grid1) and updates the output grid (grid2) accordingly. The explosion algorithm considers adjacent cells in the horizontal and vertical directions, modifying the grid based on bomb locations.

### 2.2.3. copyGrid Function:

This subroutine copies the contents of one grid (grid1) to another (grid2). It utilizes a loop to iterate through each cell, transferring the values from the source to the destination grid.

### 2.2.4. printGrid Function:

Prints the content of a given grid to the console. It uses system call 11 (print character) to display each character in the grid. The function includes newline characters to format the output grid.

## **2.3. Program Flow:**

### **Initialization:**

User provides row size, column size, seconds, and initial grid.

grid1 is initially filled with user-provided data.

grid2 is initialized as an empty grid and then grid1 is copied to the grid2 to implement functions on it.

### **Simulation Loop:**

The program enters a loop that iterates from 1 to the specified number of seconds.

In each iteration:

bomberMan function fills empty spaces in grid2 with bombs.

The current state of the grid (grid2) is printed.

explodeBombs function simulates bomb explosions, updating grid2.

The updated grid is printed again.

Finalize the loop by copying the last state of grid2 to grid1.

### ***3. Algorithm:***

The Bomberman program uses a straightforward algorithm for bomb placement and explosion. The `bomberMan` function iterates through each cell, placing bombs where the cell is empty. The `explodeBombs` function simulates the explosion by checking adjacent cells and updating the grid accordingly.

### ***4. Summary:***

The program follows a sequential flow, starting with user input and progressing through iterations of bomb placement and explosion. The use of two grids (`grid1` and `grid2`) facilitates a smooth transition between states. Key functions (`bomberMan`, `explodeBombs`, `copyGrid`, and `printGrid`) are responsible for specific aspects of the simulation. The algorithm employed is intuitive, making the program comprehensible and effective.

## 5. Flowchart

