PROJECT 1 - EMİRCAN DEMİREL- 1901042674

Program Flow:

1. Users enter the number of rows and columns as inputs for the grid.
2. The program validates inputs to ensure it falls within the allowed limits (32x32).
3. The grid is initialized using the initializeGrid function.
4. The initial grid is displayed.
5. Bombs are planted on a gridNew and copied to finalGrid then the gridNew is displayed.
6. Bombs are detonated on the finalGrid, and the finalGrid is displayed.
7. The program exits.

Data Segment:

The program begins with a data segment where various variables and constants are declared. These include:

- input_buffer: A buffer for user input.
- grid, gridNew, and finalGrid: Memory spaces allocated for the grid structures.
- dx and dy: Arrays to store direction values for rows and columns.
- Various prompts and error messages for user interaction.

Subroutines:

- *initializeGrid:* Initializes the grid with user-provided data while validating input. It takes two parameters: the number of rows in the grid ($a0) and the number of columns in the grid ($a1). It reads rows of characters from the user, validates each character ('O' or '.'), and stores them in the grid while ensuring the input does not exceed the grid dimensions. Invalid characters are replaced with '.'.
- *printGrid:* Prints the grid to the console. It takes three parameters: the base address of the grid to be printed ($a0), the number of rows in the grid ($a1), and the number of columns in the grid ($a2). It iterates through the rows and columns of the grid, loading characters from memory and printing them to the console. It uses nested loops to traverse the grid and newline characters to separate rows. This subroutine allows for the flexible display of grids of varying sizes.
- *plantBombs:* Initializes the grid with bombs and copies the bomb placement to the final grid. Uses the global variables and constants to access grid-related data. It uses nested loops to traverse the grid and plants bombs ('O') in the gridNew while also copying the bomb placement to the finalGrid. This subroutine ensures that the gridNew and finalGrid are correctly populated with bombs.
- *detonateBombs:* Detonates the bombs, updates the final grid, and clears adjacent cells. It takes two parameters: the number of rows in the grid ($a1) and the number of columns in the grid ($a2). It employs nested loops to traverse the grid, identifies bombs ('O'), detonates them by updating the corresponding cell in the finalGrid to '.', and clears adjacent cells if they exist within the grid boundaries. It uses direction arrays (dx and dy) to determine adjacent cell positions.