**SCREENSHOTS**

**A black background with white text

Description automatically generated**

**QUESTIONS**

**1) How the CURSES library is being used in this code?**

The **curses** library is used in this program to manage the terminal interface, allowing for an interactive, text-based user interface. Here's a breakdown of how it's used:

**Terminal Setup:**

* **initscr()**initializes the library and sets up the terminal in a mode where it can handle keypresses and screen manipulation.
* **cbreak()** disables input buffering, so characters are passed to the program immediately rather than waiting for a newline.
* **nodelay(stdscr, true)** makes **getch()** non-blocking, meaning it will return immediately even if no key is pressed.
* **noecho()** disables the automatic display of typed characters.
* **intrflush(stdscr, false)** disables automatic flushing of the input buffer on interrupts.
* **keypad(stdscr, true)** enables special keys (like arrow keys) to be recognized by the program.
* **curs\_set(0)** hides the cursor during the program's execution.

**Screen Output:**

* **clear()** clears the screen at the start of each loop iteration.
* **mvprintw(y, x, "text")** prints the current time at the specified coordinates (x, y) on the terminal.
* **refresh()** updates the screen to show any changes made with mvprintw() or other screen-modifying functions.

**Terminal Teardown:**

* **endwin()** deinitializes the curses library, restoring the terminal to its normal state when the program exits.

**2) From your observation, identify all the CURSES methods/functions that have been used in each program.**

* **initscr()**: Initializes the curses library and sets up the screen.
* **cbreak()**: Disables input buffering, making input available immediately.
* **nodelay(stdscr, true)**: Makes getch() non-blocking.
* **noecho()**: Prevents typed characters from being echoed to the screen.
* **intrflush(stdscr, false)**: Disables input flushing on interrupt.
* **keypad(stdscr, true)**: Enables recognition of special keys like the arrow keys.
* **curs\_set(0)**: Hides the cursor.
* **clear()**: Clears the screen for fresh drawing.
* **mvprintw(y, x, "text")**: Moves the cursor to (y, x) and prints the specified string.
* **refresh()**: Updates the screen to reflect changes made with mvprintw() or other methods.
* **endwin()**: Deinitializes the curses library and restores terminal settings.

**3) What is the difference between CURSES.h and nCURSES.h?**

**CURSES.h**: This is an older and less commonly used header for the curses library. It was historically used on older UNIX systems and doesn't support some modern features and optimizations found in ncurses.h. It is generally less portable and is considered outdated.

**nCURSES.h**: This is the more modern and widely used header for the curses library. ncurses stands for "new curses" and is an extended, enhanced version of the original curses library. It offers more features, such as improved screen handling, extended key support, and better portability across different UNIX-like systems.

When using ncurses.h, you get access to the full set of features and improvements over the older CURSES.h header for working with terminal-based interfaces.