Switch Simulation using C Language

Research:

Sources-

Youtube, Google websites- geeksforgeeks.org, www.programiz.com, www.w3schools.com, Books- "The C Programming Language" by Kernighan and Ritchie.

Analysis:

The Smart Light Control Program is a simple console application written in C that allows users to manage a smart light. The program provides functionalities to turn the light on or off, adjust its brightness, and change its color.

Here's an analysis of its components:

1. It first prompts the user to turn the light on or off.
2. If the light is turned on, it enters a loop where the user can choose to adjust brightness, change color, or exit.
3. The program uses simple conditional statements to handle user input.

Key Features:

- User-Friendly: Clear prompts and feedback help users understand the available options and actions.
- Input Safety: The code prevents buffer overflow when changing color by limiting the input size.
- Loop Structure: The use of a while loop allows for continuous interaction until the user decides to exit.

Ideate:

Concept Applications:

Smart appliances [Home automation): To control daily appliances like lights, fan etc using binary decision making.

Purpose: To read Binary values(0/1) and give ON/OFF conditions as output resulting in switch simulation. To change brightness and color of a light by giving user menu.

Build:

Algorithm (for base code):

1. Start the program.

2. Declare an integer variable ip to store user input.

3. Display a message asking the user for input: "Give input (0/1):"

4. Read the user input and store it in the variable jp.

5. Check the value of jp:

If ip is equal to 0: Print "Switch is OFF".

Else if ip is equal to 1: Print "Switch is ON"

Else: Print "Error: Incorrect input"

6. End the program.

Base Code:

```
#include <stdio.h>

int main()

{

int ip;

printf("Give input 0/1");

scanf("%d" , &ip);

if(ip== 0)

{ printf("Switch is OFF");}

else if (ip==1)

{printf("Switch is ON");}

else

{printf("Error:Incorrect input");}

return  0;
```

}

Real Life Applications:

1.Smart appliances(Home automation): To control daily appliances like lights, fans, etc using binary decision making.

Code:

```
#include <stdio.h>

// Function to set the brightness of the light

void Brightness(int brightness)

// Check if the brightness level is valid

{

    if (brightness < 0 || brightness > 100)

      { printf("Invalid brightness level.\n"); }

    else

      { printf("Brightness set to %d%%.\n", brightness); }

}

// Function to change the color of the light

void changeColor(char *color)

{

 printf("Enter the new color of the bulb (max 19 characters): ");

scanf("%19s", color); // Read color input and prevent buffer overflow

}

int main()

{

 int choice, state, brightness; // Variables for user input

char color[20] = "White";// Initialize the color of the light

// Prompt user to turn the light ON or OFF
```

```
printf("Enter light ON/OFF:\n");
printf("Enter 1 to turn ON, 0 to turn OFF: ");
scanf("%d", &state);
// If the user wants to turn the light ON
if (state == 1)
   { printf("Lights ON.\n");
// Infinite loop for user menu until they choose to exit
    while (1)
    // Display the smart light control menu
       { printf("\nSmart Light Control Menu:\n");
         printf("1. Adjust Brightness\n");
         printf("2. Change Color\n");
         printf("3. Exit\n");
         printf("\nEnter your choice: ");
         scanf("%d", &choice); // Read user choice
  // Adjust brightness
  if (choice == 1)
    { printf("Enter brightness level (0-100): ");
      scanf("%d", &brightness); // Read brightness input
      Brightness(brightness); } // Call function to set brightness
  else if (choice == 2)
  // Change color
       { changeColor(color); // Call function to change color
         printf("The color of the bulb has been changed to: %s\n", color); }
   else if (choice == 3)
```

```
// Exit the program
        { printf("Exiting program.\n");
          return 0; } // Exit main function
   else
   // Handle invalid choice
      { printf("Invalid choice. Please try again.\n"); } } }
else
 // If the user turns the light OFF
    { printf("Lights OFF.\n"); }
return 0;
 } // Return from main function
```

Test:

This code was tested on various compilers like VS Studio, and online compilers like Programiz.com, w3schools.com. The following output was attained for different input cases.

OUTPUT-

Case1- Input is 0(switch is off)

Enter light ON/OFF:

Enter 1 to turn ON, 0 to turn OFF: 0

Lights OFF.

Case2- Input is 1(switch is on), brightness set to 50% & color set to blue.

Enter light ON/OFF:

Enter 1 to turn ON, 0 to turn OFF: 1

Lights ON.

Smart Light Control Menu:

1. Adjust Brightness

2. Change Color

3. Exit

Enter your choice: 1

Enter brightness level (0-100): 50

Brightness set to 50%.

Smart Light Control Menu:

1. Adjust Brightness

2. Change Color

3. Exit

Enter your choice: 2

Enter the new color of the bulb (max 19 characters): blue

The color of the bulb has been changed to: blue

Smart Light Control Menu:

1. Adjust Brightness

2. Change Color

3. Exit

Enter your choice: 3

Exiting program.

Implement:

1. Function Definitions:

- **Brightness Function**: This function takes an integer input representing the desired brightness level (0 to 100). It checks if the input is within this range and displays an appropriate message.
- **ChangeColor Function**: This function prompts the user to enter a new color for the light. It safely reads the input string, ensuring it does not exceed the allocated buffer size.

2. Main Function:

- The program begins by prompting the user to turn the light on or off using a simple menu.
- If the user chooses to turn the light on, the program enters an infinite loop displaying a menu with three options: adjust brightness, change color, or exit.
- Based on user input, it calls the respective functions:
    - **Adjust Brightness**: It requests the brightness level and passes it to the Brightness function.
    - **Change Color**: It calls the ChangeColor function to read and update the light's color.
    - **Exit**: It terminates the program.

3. Input Validation:

- The program includes checks to validate user inputs, ensuring that the brightness level is within the specified range and that menu choices are valid.

4. User Interaction:

- The program is designed for straightforward interaction, making it user-friendly. Clear prompts and feedback messages guide the user through the options.

Conclusion

The Smart Light Control Program serves as a basic example of how user input can be handled in a switch simulation application. It utilizes functions for modular design, ensuring that each functionality (brightness adjustment and color change) is summarized, making the code easier to read and maintain. Overall, it demonstrates fundamental programming concepts such as loops, conditionals, and user input handling in C.

Group Members:-

Sanskruti H. Gudhe :B24CE1043

Shivtej C. Wagare :B24CE1041

Sakshi A. Badakh :B24CE1045

Shubham S. Dorake :B24CE1042

Soham D. Badarkhe :B24CE1040