Evan Chen

CIS15BG

Professor Lee-Klawender

Programming HW#5 Design

Program Name: ColorStruct.c

**Step 1: Analyze the Problem**

**Output**: Displays color number (in decimal & hexadecimal)

**Input**: Reads a list of color names and color number values

**Step 2: Design (Structure Chart, Pseudo code, Algorithmic pattern, Modular Specs)**

**Step 3: Main Pseudocode:**

CALL readInputColor(COLOR)

CALL colorSort(COLOR,numElems)

CALL displayColor(COLOR,numElems)

CALL userQuery(COLOR,numElems)

CALL freeMem

end main

**Step 4: Logic of program**:

This program is much like Program#4. However, it uses a typedef struct called COLOR to hold the colorName and colorValue now. The program will first open a file and if opened correctly, each line will be read into a temporary string. As each line is read, the temporary string and address of the current element of the colorName in COLOR will be passed to another new function. The new function will fill up the colorName and colorValue and dynamically allocated memory for colorName in COLOR. Afterwards, this program will sort the arrays in COLOR by color name by using another function. The sorted COLOR will then be displayed and the user will be prompted to insert a color name. Another program will search COLOR and display the decimal and hexadecimal value corresponding to the color name entered. When the user enters quit, the program will free all dynamically allocated memory and close.

**Step 5: Module Specs**

**1. openFile**

Return Value: FILE\*

Receives: prompt(string), mode(string)

Reference Parameters: prompt(string), mode(string)

Preconditions: None

Logic: Prompts user to provide the file’s name. The function will then read the file name into a local string. If the name of the file is QUIT, then the function returns NULL and the program quits. Otherwise, the function will open this file with the received by the mode(string) and return the file as a FILE\*

**2. readInputColor**

Return Value: numElems(int)

Receives: COLOR(typedef struct)

Reference Parameters: colorName and colorValue in COLOR

Preconditions: COLOR has been defined

Logic: This function will call openFile to open a file. If it opens the strings will be read into a temporary string. The struct COLOR contain char\*colorName and int colorValue in it. The temporary string and COLOR will be passed to function #3. This function will then return the # elements of the file or 0 if the file was not opened.

**3. fillColorStruct**

Return Value: void

Receives: tempStr(string),COLOR(struct) containing char\*colorName and int colorValue

Reference Parameters: colorName and colorValue (in typedef struct COLOR)

Preconditions: COLOR is defined with colorName and colorValue in it both initialized. tempStr is declared and initialized

Logic: This function will first dynamically allocate memory for colorName located in the struct COLOR. It will then parse and read tempStr into the newly allocated memory. The numbers following each color name will be converted into ints and stored into colorValue

**4. colorSort**

Return Value: void

Receives: COLOR(typedef struct), numElems(int)

Reference Parameters: colorName and colorValue(in typedef struct COLOR)

Preconditions: COLOR is defined with colorName and colorValue in it are both initialized, numElems declared and initialized

Logic: This function will sort colorName and its respective colorValue (both located in COLOR).

**5. displayColor**

Return Value: void

Receives: COLOR(typedef struct), numElems(int)

Reference Parameters: None

Preconditions: This function will display the colorName and decimal & hexadecimal of colorValue to stdout. The data will be lined up in columns and numbers right justified

**6. freeMem**

Return Value: void

Receives: colorName(char\* located in typedef struct COLOR)

Reference Parameters: none

Preconditions: COLOR defined and colorName has been initialized

Logic: This function will free all dynamically allocated memory.

**7. strlCmp**

Return Value: compare(int)

Receives: str1(string),str2(string) both located in COLOR

Reference Parameters: str1,str2

Preconditions: str1, str2 are pre-declared and initialized

Logic: This function will compare 2 strings with strlcmp which ignores the case and will return an int back based on if str1 and str2 equal each other or if one is greater or less than the other.

**8. userQuery**

Return Value: void

Receives: COLOR(typedef struct),numElems(int)

Reference Parameters: None

Preconditions: COLOR is defined and colorName & colorValue inside are initialized, numElems declared and initialized

Logic: Prompts user for a color name and calls another function which searches for it in colorName. When found the colorName’s respective colorValue will be displayed in decimal and hexadecimal format. If not found, it will let the user know. This function will continue to prompt and display until user inputs Quit.

**9. searchColor**

Return Value: index(pointer to COLOR struct)

Receives: COLOR(typedef struct), colorname(string), numElems(int)

Reference Parameters: None

Preconditions: COLOR is defined and colorName & colorValue inside are initialized, numElems declared and initialized

Logic: This will search COLOR for the colorname parameter a return a pointer to the COLOR struct where found. It will return NULL if it is not found.