Part 1:

Functional Requirements:  Rewrite Chapter 5 Programming Assignment so that it (1) changes the games AND (2) utilizes functions.  You will need at least 4 functions that are called by main: menu, displayBackwards,  colors, and specialChar.

The functions operate as follows:

int menu() displays the menu and prompts for the user's choice and returns that choice.

void displayBackwards() asks the user to input an upper-case alphabetic character between M and R.  Validate that input is an upper-case alpha between M and R.  Then display all the letters backwards from that letter down to 'A'.  For example, if the user enters, 'M', then the program would display M L K J I H G F E D C B A

void colors() asks the user to input two primary colors (red, blue, or yellow).  Validate that inputs are primary colors.  Then display the two colors with their secondary color.  For example, if the user enters "red" and "blue" then the program would display red \*\*purple\*\*blue.

Chart, bubble chart

Description automatically generated

void specialChar() prompts the user for a positive integer and one special character from this list: left curly bracket, ampersand, percent, pound, asterisk. Validate for positive integer.   Validate for allowable special character.  Then display the special character as many times as the integer indicates.  For example if the user enters 5 for the integer and & for the special character, then the program would display &&&&&

Programming Requirements:

Each function should have it's own header block.  It should state the name of the function, the purpose of the function and what it is passed or what it returns.  These header blocks are created using comment lines within your code.

Rule #1:  Absolutely no global variables.  If main() needs to know what choice the user made from the menu, then the function menu() returns that value.  No global variables under any circumstances.

Input validation is as explained above.  It should always be constructed with a while loop.  While the input is not valid, display error message and prompt again for input.

Pseudocode for the main function might look like this:  (I underlined a few side comments for you - but they are not part of the p-code.)

*declare your functions*

*main()*

*int choice;*

*do*

*choice =  displayMenu (this function will return the user's choice)*

*switch on choice*

*case 1*

*call function to handle displaying backwards*

*case 2*

*call function to handle displaying colors*

*case 3*

*call function to handle special characters*

*case 4*

*output goodbye statement*

*default*

*handle incorrect choice (ie, choice not 1-4)*

*end switch*

*while (choice not equal to 4)*

*end main*

*--------pseudocode for displayMenu function*

*int function displayMenu()*

*int userChoice                                       //this userChoice is local to displayMenu function*

*display the menu*

*input the userChoice*

*return userChoice                                  //here userChoice is being returned to main()*

*------------*

*Special Note:  If you change to functions but do not change the games, you will receive 0 points.  If you change the games but do not switch to using functions, you will receive 0 points.*

Part 2:

Today we are going to create a program for my photography business.  Here are your requirements:

Customer Number:  (integer >0)

Category:  (wedding, dog portrait, action image)

Hours worked:  (float >0)

Number of 8 X 10 prints requested (int > = 0)

Number of enlargements requested (int >=0)

Formulas you need:

Wholesale Price = (hours worked \* 50 + prints \* 15 + enlargements \* 35)

Retail Price = Wholesale Price \* 5;

Programming Requirements:

1.  Main must call one separate function for each item input.  That function will validate the amount and then *return* the validated amount.

2.  Then main should call  just one function to calculate both wholesale and retail prices.  This function will pass the hours, prints and enlargements by value and it will use reference parameters to receive the two calculated amounts.

3.  Finally, main will pass all the input data and the two calculated price to a function that will display everything in a professional manner.  Make money look like money and keep your output aligned.

Pseudocode:

*prototype functions*

*main*

*call inputCustomerNumber to input and validate customer number*

*call inputCategory to input and validate category*

*call inputHours to input and validate hours worked*

*call inputPrints to input and validate 8X10 prints requested*

*call inputEnlargements to input and validate 14X20 prints requested*

*call calcPrices to calculate wholesale and retail prices*

*call displayEverything to display all inputs and results*

*end main*

*-------------*

*int inputCustomerNumber()*

*display prompt*

*input item number*

*while(item <=0)*

*display error message*

*input item number*

*end while*

*return customerNumber*

*end function*

*------the rest of the input functions are similar*

*void calcPrices( float hours, float prints, float enlargements, float &wholesale, float & retail)  \*\*the last two are pass by ref*

*-----the formula goes here*

*void display(.....)*

*----pass it the input data AND the two prices and it displays them in a professional manner.*