# GNG1106: Solutions for Homework 2

## QUESTION 1

(2 marks total. Correct explanation 1.5 marks – correct results 0.5 marks)

This is a simple block of code and basically what happens here is that first variable **i** of type **int** with value **0** is initiated. Then the program enters an **infinite while loop** because of condition 1 (remember that any non-zero value is evaluated as 'true'). Therefore the while loop keeps on executing the statements within its block until it reaches the **Break statement**. Therefore inside of the loop we have:

1. The value of **i** is **incremented** 1 unit;
2. The value of **i** is **printed**.
3. And if the value of **i > 3**, **the loop is terminated**.

The loop will iterate four times in total until **i** reaches number 4 and 4 is printed. Afterwards the condition of the "***if statement***" in line 9 is evaluated as true **(i > 3)**. As a result the break statement in line 10 gets executed and the infinite loop is terminated. The following output is printed:

1
2
3
4

```
1   #include <stdio.h>
2
3   int main() {
4       int i=0;
5       while(1)
6       {
7           i=i+1;
8           printf("%d\n", i);
9           if (i>3)
10              break;
11      }
12  }
13
14  C:\Users\workspace\codeBlocks\Homework2-1\bin\Debug\Homework2-1.exe
15  1
16  2
17  3
18  4
19
20  Process returned 2 (0x2)   execution time : 0.206 s
    Press any key to continue.
```

*Figure 1 - Question 1 results*

# QUESTION 2

(2 marks total. Correct explanation 1.5 marks – correct results 0.5 marks)

The code compiles and runs and the output of the program is be the following:

4
3
2
1
0

In this block of code we are again dealing with a while loop. But the difference is that the loop condition here is the evaluation of variable **i**. Remembering from previous homework and lecture notes, as long as the value of variable is non-zero, it is considered as '**true'**.

Hence, we start with the initial value of **i = 5**. In every iteration that value is decreased 1 unit and then printed until we reach the value **0.** That's when the while loop ends.

Keep in mind in the last loop in which we have "**while(1)"** (i=1) the number "**0"** is printed because the **statement "i = i – 1" is taking place before the print statement** within the **while block.** Then the next **while(0)** evaluation takes place in which the while loop terminates (loop body is skipped) with five iterations in total.

```
1   #include <stdio.h>
2
3   int main() {
4       int i=5;
5       //. . .
6       while(i)
7       {
8           i=i-1;
9           printf("%d\n", i);
10      }
11  }
12
13  C:\Users\workspace\codeBlocks\Homework2-1\bin\Debug\Homework2-1.exe
14  4
15  3
16  2
17  1
18  0
19
    Process returned 2 (0x2)   execution time : 0.029 s
    Press any key to continue.
```

*Figure 2 - Question 2 results*

# QUESTION 3

a) Programming style (indentation, commenting, naming convention, etc) : 10% → 0.6 marks
b) Problem solving and design logic (namely, things you do in Step 1, 2, 3) : 30% → 1.8 marks
c) Correct use of syntax : 20% → 1.2 marks
d) Correct execution and results : 30% → 1.8 marks
e) The presentation of lab report : 10% → 0.6 marks

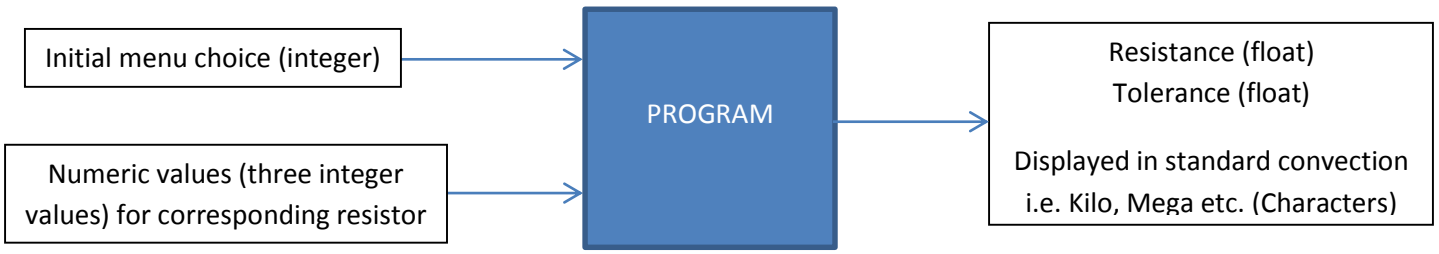
Lack of a lab report (i.e if only C code is submitted) you will lose marks for (b) and (e).

**STEP 1: Problem statement:**

Design a program that would accept the four bands of color on a resistor and would calculate its resistance value. The program would provide an options menu to enter the values, seek help on how to provide these values or exit the program

**STEP 2: I/O description**

- **Inputs**
  1. **The menu value** (integer) with three choices.
     a. '1' to calculate the resistor value
     b. '2' to read a help text on using the program.
     c. '3' to exit.
  2. **Colour code** (If the user chooses 1 as menu value)
     a. An integer between [0,99]
     b. An integer between [0,9]
     c. An integer number: either of -1 or -2
- **Output**
  - The **resistance** and the **tolerance** printed in characters (standard conventions: 'Kilo', 'Mega', 'Giga' etc. should be used)



**STEP 3: Test cases and Algorithm Design**

To compute the resistance value of the resistors based on their band colors, we should first match the colors on the resistor to the appropriate digits based on the following table:

| Color | 1st band | 2nd band | 3rd band (M, multiplier) | 4th band (T, tolerance) |
|---|---|---|---|---|
| Black | 0 | 0 | $\times 10^0$ | |
| Brown | 1 | 1 | $\times 10^1$ | |
| Red | 2 | 2 | $\times 10^2$ | |
| Orange | 3 | 3 | $\times 10^3$ | |
| Yellow | 4 | 4 | $\times 10^4$ | |
| Green | 5 | 5 | $\times 10^5$ | |
| Blue | 6 | 6 | $\times 10^6$ | |
| Violet | 7 | 7 | $\times 10^7$ | |
| Gray | 8 | 8 | $\times 10^8$ | |
| White | 9 | 9 | $\times 10^9$ | |
| Gold | | | $\times 10^{-1}$ | ±5% |
| Silver | | | $\times 10^{-2}$ | ±10% |

*Figure 3- Resistor color code table*

After acquiring the 4 digits ($D_1$, $D_2$, $D_3$ and $D_4$), the following formula should be used to calculate the resistance value followed by tolerance value:

- $Resistance = D_1D_2 \times 10^{D_3}$
- $Tolerance =$
  - $if\ (D_1 = -1) \rightarrow Tolerance = 5\%\ (Resistance)$
  - $if\ (D_1 = -2) \rightarrow Tolerance = 10\%\ (Resistance)$
- $\boldsymbol{Result}$: " $\boldsymbol{Resistance \pm Tolerance}\%$ "

To display the flowchart of this program we divided it into one main flowchart and multiple smaller flowcharts:

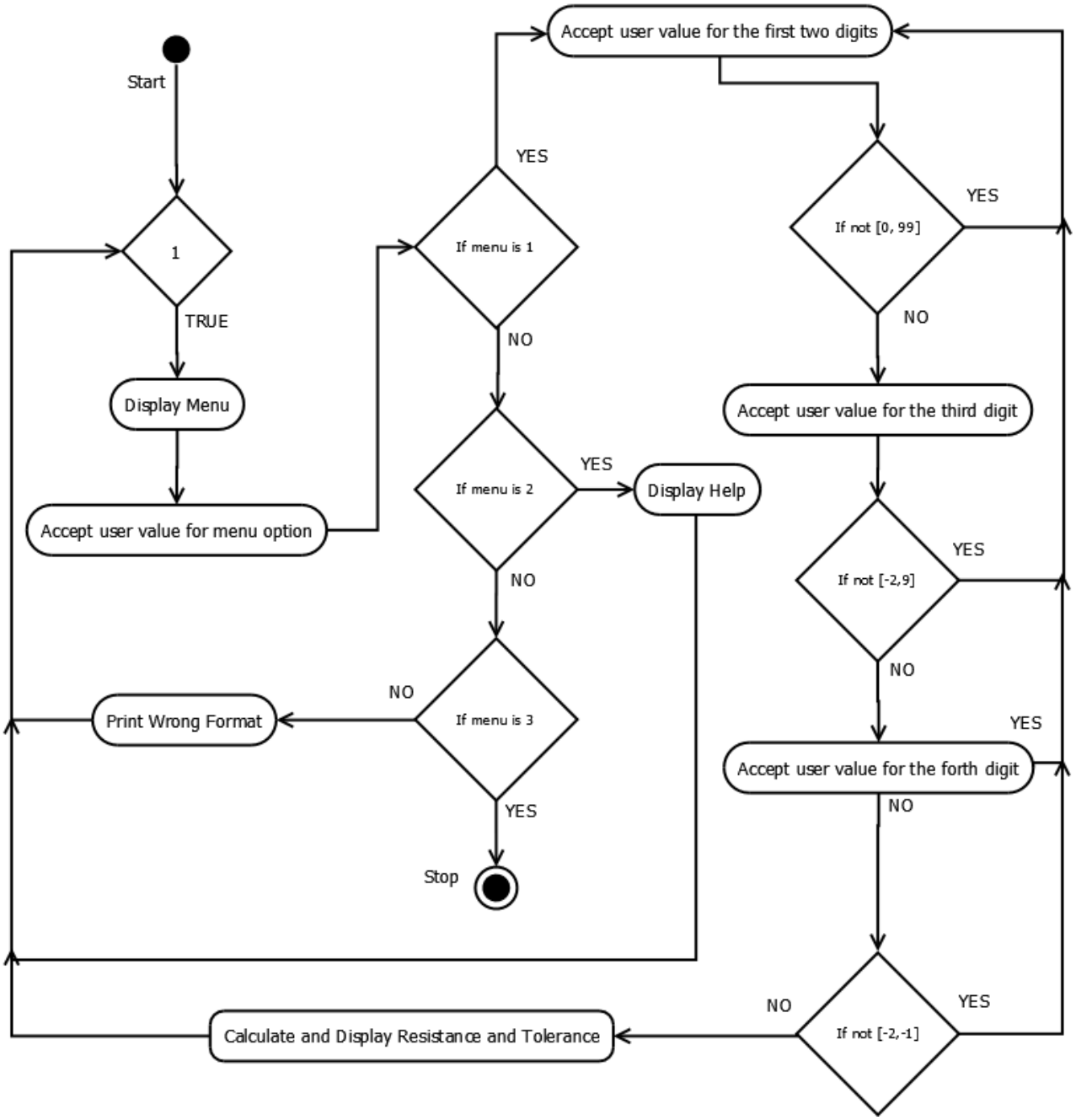



*Figure 4 - Main Program Flowchart*

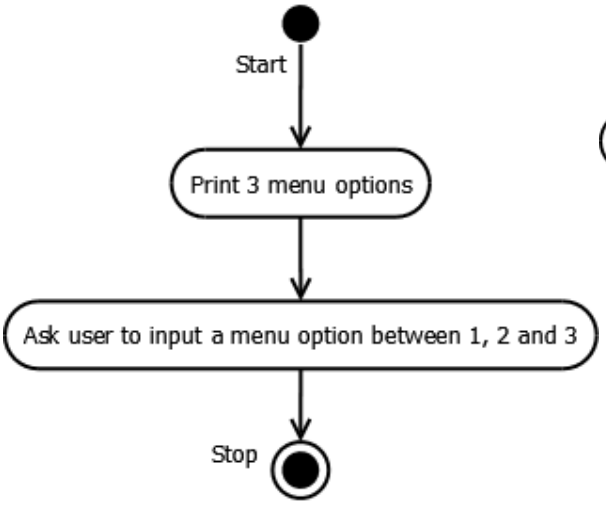



*Figure 5 - Display Options Module*

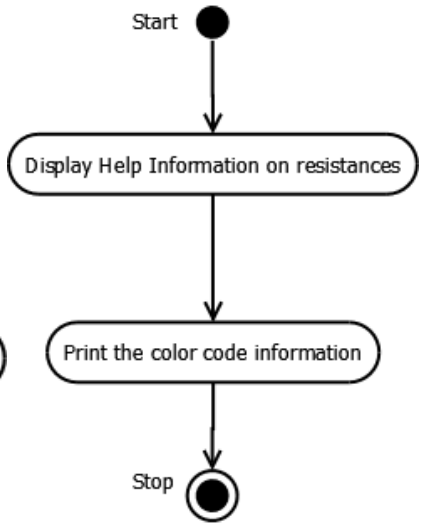



*Figure 6 - Display Options Module*

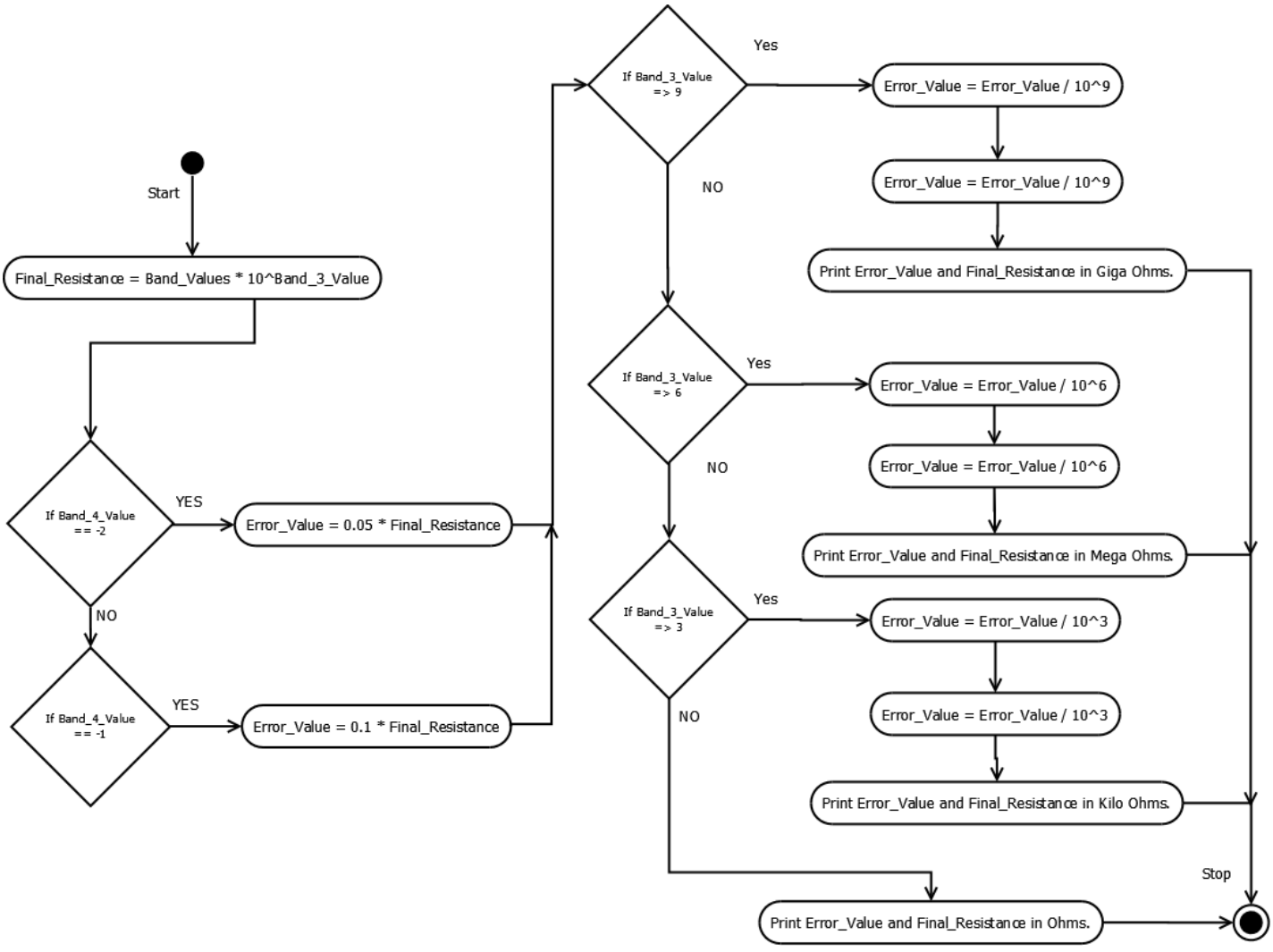



*Figure 7 - Resistance and Tolerance Display Module*

**STEP 4: C Code for the program**

While the following code has been formatted to be easier to read, it is advised to use caution while copying the code onto CODE::Blocks. Some statements may inadvertently give rise to extra line spacing, etc. in between them and might make the program hard to use while on the IDE.

We will try to upload the .C file as well so that you can download that and run it directly and use the following code just for reading purposes.

*Note – This program can be done using strings or characters too. The user can enter the colours and the program can assign values based on the colours, but we have avoided that method here as they have not been taught in class as of now.*

```
/*-------------------------------------------------
Name: yournamehere, Student Number: 0000001
Date: October 6, 2014.
Program: resistance.c

Description: This program converts color bands
information on a resistor to its resistance
value in Ohm.


* In the final output display, %.2f has been used
in place of %f.

This is simply to make the output look better
by removing unnecessary decimals. %.2f
essentially truncates the decimals to 2 digit
after the decimal point. Without it, the
compiler will print (for example) 2.1000 Kilo Ohm.

Using it will display the result as 2.10 Kilo Ohm.
Two has been used because the lowest amount of
tolerance is 0.05 and that's the largest precision
the program requires.
-------------------------------------------------*/


#include <stdio.h>
#include <stdlib.h>
#include <math.h>


int main()
{
    int menu_code;
    int band_values, band_3_value, band_4_value;
    float final_resistance, error_value;

    // Keep printing the menu unless teh user decides to exit.
    while (1) {
        // printing the menu
        printf("Please choose an option:\n\n");
        printf("1. Enter \"1\" to calculate resistance value\n");
```

```
        printf("2. Enter \"2\" to Help\n");
        printf("3. Enter \"3\" to Exit\n\n");

        // getting the user input
        scanf("%d",&menu_code);
        printf("\nYour option is: %d\n\n", menu_code);

        // Analyzing the user selected option; if user want to calculate
        if(menu_code == 1) {
            printf("\t\t\t****** Calculate The Resistance ******\n\n");
            while (1)
            {
                printf("\nPlease enter the first two significant digits of
the resistance value.\n\n");
                printf("Choose a number between [0,99] corresponding to these
colors:\n");
                printf("Black = 0\nBrown = 1\nRed = 2\nOrange = 3\nYellow =
4\n");
                printf("Green = 5\nBlue = 6\nViolet = 7\nGray = 8\nWhite=
9\n\n");
                scanf("%d", &band_values);

                if (band_values < 0 || band_values > 99)
                {
                    printf("\nOnly numbers between [0,99] are accepted,
please try again.\n");
                    continue;
                }
                printf("\nPlease enter the number between [-2,9]
corresponding to third band color of the resistor:\n\n");
                printf("Silver = -2\nGold = -1\nBlack = 0\nBrown = 1\nRed =
2\nOrange = 3\n");
                printf("Yellow = 4\nGreen = 5\nBlue = 6\nViolet = 7\nGray =
8\nWhite= 9\n\n");
                scanf("%d", &band_3_value);


                if (band_3_value < -2 || band_3_value > 9)
                {
                    printf("\nOnly numbers between [-2,9] are accepted,
please try again.\n");
                    continue;
                }

                printf("\nPlease enter number that corresponds to the fourth
color band of the resistor:\n\n");
                printf("Silver = -2\nGold = -1");
                scanf("%d", &band_4_value);

                if (band_4_value < -2 || band_4_value > -1)
                {
                    printf("\nOnly numbers between [-2,-1] are accepted,
please try again.\n");
                    continue;
                }

                // eveything good up to this point; exit the while loop
                break;
            }
```

```
            // 2.calculating the multiplier value
            final_resistance = band_values * pow(10, band_3_value);

            //3. Calculating the tolerance value.
            if (band_4_value == -1)
            {
                error_value = 0.05 * final_resistance;
            }
            else if (band_4_value == -2)
            {
                error_value = 0.1 * final_resistance;
            }

            // 4, Printing the final resistance value.
            // %.2f is used instead of %f simply for more readable output.
            if(band_3_value >= 9) {
                final_resistance = final_resistance / pow(10, 9);
                error_value = error_value / pow(10, 9);
                printf("\nThe resistance value is %.2f ~ %.2f Giga
Ohms.\n\n", final_resistance, error_value);
            }
            else if (band_3_value >= 6) {
                final_resistance = final_resistance / pow(10, 6);
                error_value = error_value / pow(10, 6);
                printf("\nThe resistance value is %.2f ~ %.2f Mega
Ohms.\n\n", final_resistance, error_value);
            }
            else if (band_3_value >= 3) {
                final_resistance = final_resistance / pow(10, 3);
                error_value = error_value / pow(10, 3);
                printf("\nThe resistance value is %.2f ~ %.2f Kilo
Ohms.\n\n", final_resistance, error_value);
            }
            else {
                printf("\nThe resistance value is %.2f ~ %.2f Ohms.\n\n",
final_resistance, error_value);
            }

            // Help menu
        } else if (menu_code == 2) {
            printf("\t\t\t****** HELP ******\n\n");

            printf("A Resistor is a circuit element designed to have
electrical resistance\n");
            printf("between its ends. Resistance values are expressed in
ohms. \n\n");

            printf("Resistance values are usually marked with coloRed bands
that encode \n");
            printf("their resistance values.\n\n");

            printf("The first two bands encode the first two significant
digits of the \n");
            printf("resistance value, the third is a power-of-ten multiplier,
and the \n");
            printf("fourth is the tolerance accuracy of the value.\n\n");

            printf("This program will calculate the value of a resistor based
on its color code.\n");
```

```
            printf("The color code values are presented in the following
table:\n\n");

            printf("Color  \t\t1st band \t2nd band \t3rd band \t4th band\n");
            printf("Black  \t\t0 \t\t0 \t\t10^0 \n");
            printf("Brown  \t\t1 \t\t1 \t\t10^1 \n");
            printf("Red    \t\t2 \t\t2 \t\t10^2 \n");
            printf("Orange \t\t3 \t\t3 \t\t10^3 \n");
            printf("Yellow \t\t4 \t\t4 \t\t10^4 \n");
            printf("Green  \t\t5 \t\t5 \t\t10^5 \n");
            printf("Blue   \t\t6 \t\t6 \t\t10^6 \n");
            printf("Violet \t\t7 \t\t7 \t\t10^7 \n");
            printf("Gray   \t\t8 \t\t8 \t\t10^8 \n");
            printf("White  \t\t9 \t\t9 \t\t10^9 \n");
            printf("Gold   \t\t  \t\t  \t\t10^-1 \t\t~ 5%% \n");
            printf("Silver \t\t  \t\t  \t\t10^-2 \t\t~ 10%% \n\n");

            printf("For example, if the first band is Green, the second is
Black, the \n");
            printf("third band is Orange, and the fourth one is Silver, the
resistor has \n");
            printf("a value of 50 * 10^3 ~ 0.1 * 50 * 10^3 ohm, or 50 ~ 5
kilo ohm. \n\n");

            // Exit option menu
        } else if(menu_code == 3) {
            printf("****** Exiting program.. ******\n", menu_code);
            return 0;
        } else {
            // when user selects a value other than 1, 2 or 3:
            printf("Excuse me, I did not get that.\n", menu_code);
        }
    }

    return 0;
}
```

**STEP 5: Sample output for the above code**

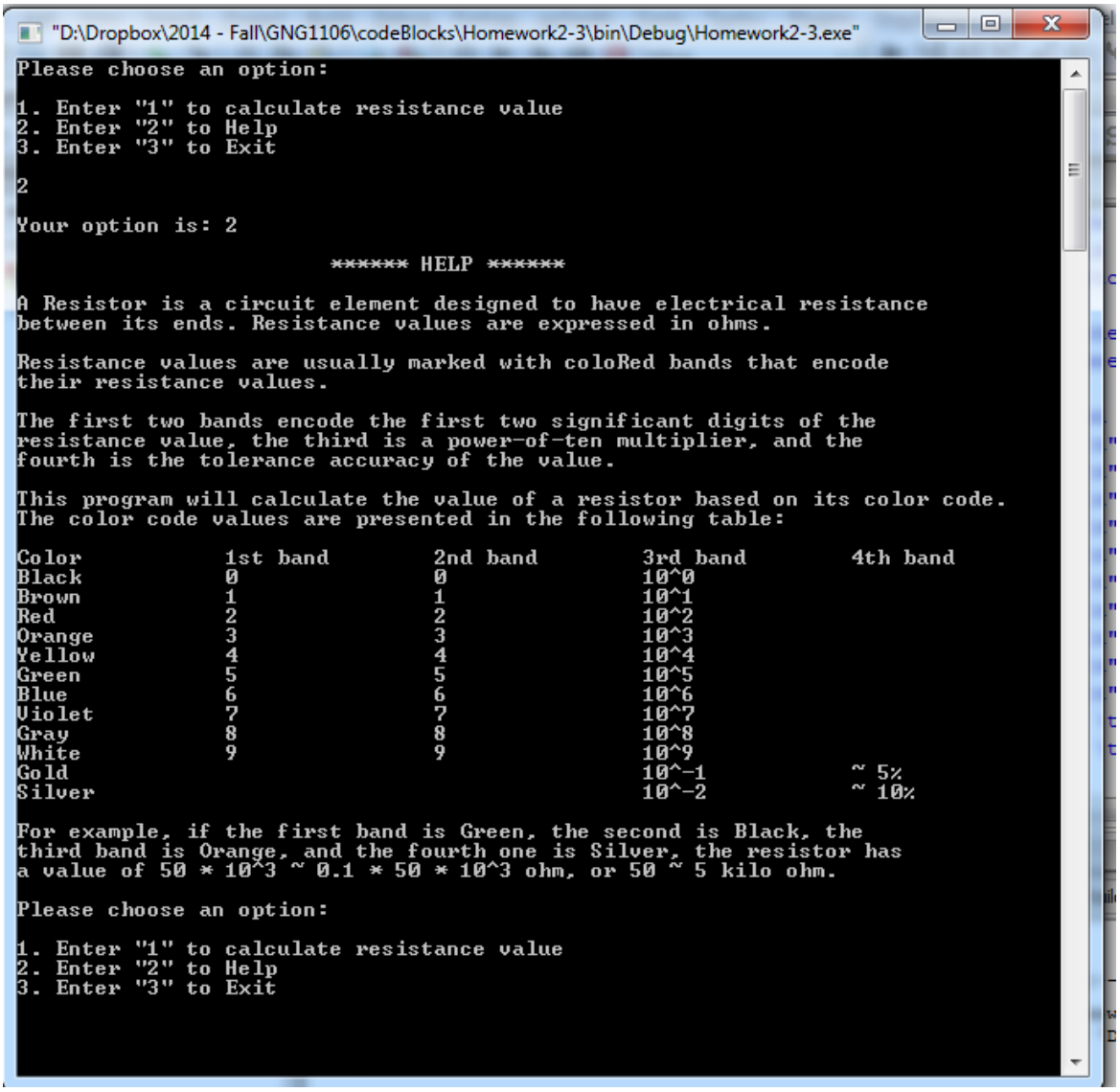



*Figure 8 - Test 01 Showing Help Menu*

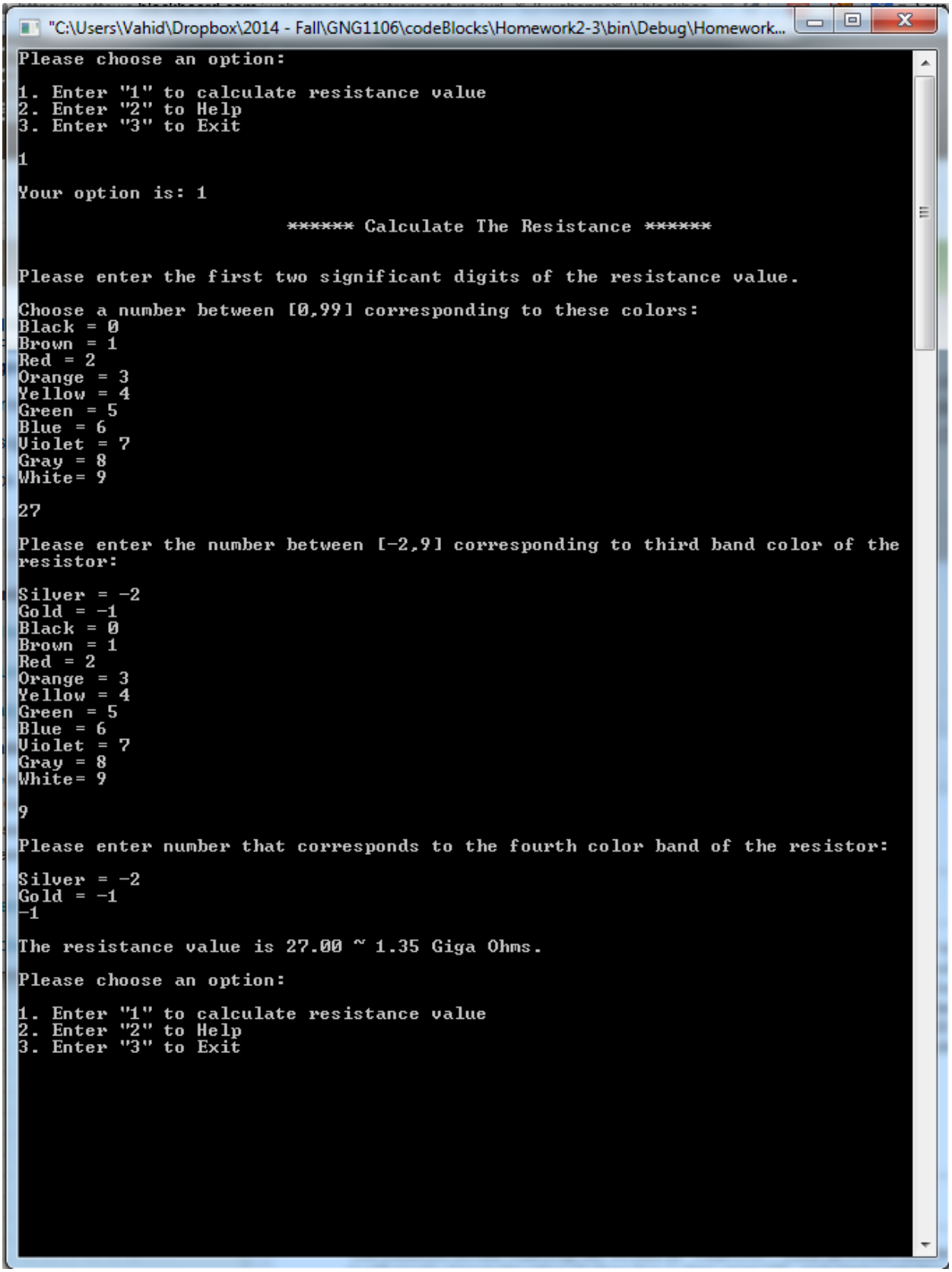



*Figure 9 - Test02 A successful result*

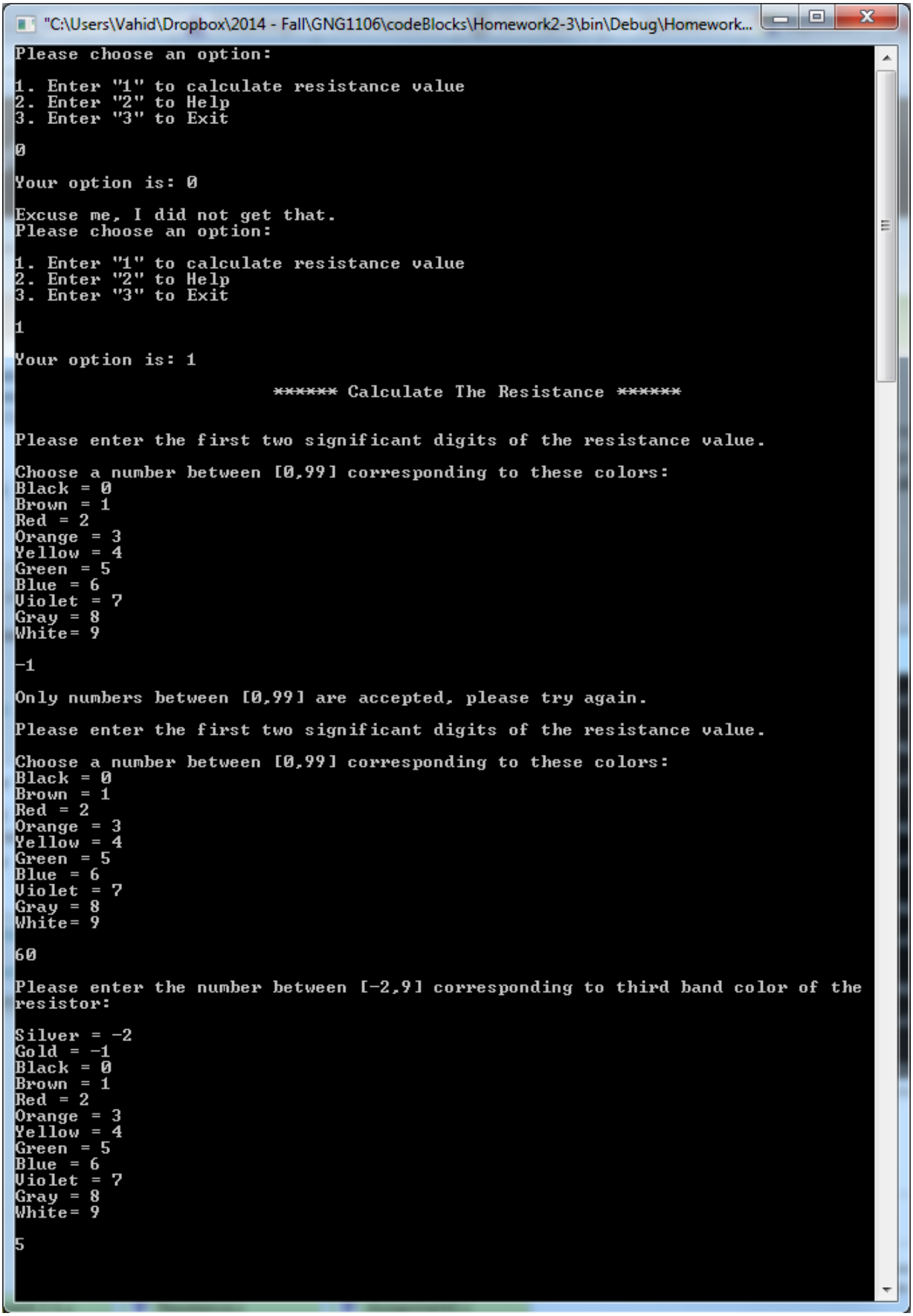



*Figure 10 - Testing some invalid inputs*