만도 SW 과제

정보통신소프트웨어학과

1기 201732046 장영하

제출일 2022/03/10

-내가 짠 코드

#include <stdio.h>

#include <math.h>

int main()

{

int band;

char c\_band[5]={0};

int i = 0;

long long rs\_max = 0, rs\_min = 0, rs = 0;

printf("색코드 숫자 입력 : ");

scanf("%d", &band);

for (i = 0; i < band; i++)

{

printf("%d번째 색을 입력해주세요 ( 검정~흰색 (a ~ j), 금색 은색 무색 ( G, S, N ) ) : ", i + 1);

scanf(" %c", &c\_band[i]);

switch (c\_band[i])

{

// 색상코드

case 'a': //검

c\_band[i] = 0;

break;

case 'b': //갈

c\_band[i] = 1;

break;

case 'c': //빨

c\_band[i] = 2;

break;

case 'd': //주

c\_band[i] = 3;

break;

case 'e': //노

c\_band[i] = 4;

break;

case 'f': //초

c\_band[i] = 5;

break;

case 'g': // 파

c\_band[i] = 6;

break;

case 'h': // 보

c\_band[i] = 7;

break;

case 'i': // 회

c\_band[i] = 8;

break;

case 'j': // 흰

c\_band[i] = 9;

break;

//색상코드 오차값

case 'G':

case 'S':

case 'N':

break;

//잘못 입력했을때 종료

default:

printf("입력 가능한 값이 아닙니다.");

return 0;

}

}

if (band == 4)

{

rs = ( (c\_band[0] \* 10) + c\_band[1] ) \* pow(10, c\_band[2]);

//오차

switch (c\_band[3]) {

case 'S':

rs\_min = rs \* 0.9;

rs\_max = rs \* 1.1;

break;

case 'G':

rs\_min = rs \* 0.95;

rs\_max = rs \* 1.15;

break;

case 'N':

rs\_min = rs \* 0.8;

rs\_max = rs \* 1.2;

break;

}

printf("저항값 : %lld \n", rs);

printf("저항 최소값 : %lld \n", rs\_min);

printf("저항 최대값 : %lld \n", rs\_max);

}

else if (band == 5)

{

rs = ( (c\_band[0] \* 100) + (c\_band[1] \* 10) + c\_band[2] ) \* pow(10, c\_band[3]);

//오차

switch (c\_band[4]) {

case 'S':

rs\_min = rs \* 0.9;

rs\_max = rs \* 1.1;

break;

case 'G':

rs\_min = rs \* 0.95;

rs\_max = rs \* 1.15;

break;

case 'N':

rs\_min = rs \* 0.8;

rs\_max = rs \* 1.2;

break;

}

printf("저항값 : %lld \n", rs);

printf("저항 최소값 : %lld \n", rs\_min);

printf("저항 최대값 : %lld \n", rs\_max);

}

else

printf("색코드가 4 또는 5가 아닙니다\n");

return 0;

}

* 숙제 코드

#include <stdio.h>

#include <math.h>

int Resistor\_value[5] = { -1, };

char Resistor\_color[13] = { 'B','b','r','o','y','g','u','p','g','w','G','S','N' };

// 0 1 2 3 4 5 6 7 8 9 10 11 12 No color = N

int main(void)

{

int no\_band = 0; // 반드시 변수 선언과 함께 초기화 할 것

int i = 0;

char temp\_color = ' ';

int resistor\_value = -1;

int decimal = 0;

int tolerance = 0;

printf("----------------------------------------------\n");

printf("------- Resistor Color code Conversion -------\n");

printf("----------------------------------------------\n");

do

{

printf(" Input number of reisitor band (4,5) : ");

scanf("%d", &no\_band);

printf("\n");

} while ((no\_band < 4) || (no\_band > 5));

printf(" no\_band : %d \n\n", no\_band);

for (i = 0; i < no\_band - 1; i++)

{

printf("%d 번째 칼라를 입력하세요. ", i + 1);

scanf("%\*c%c", &temp\_color); // \n을 지우는 방법

for (int j = 0; j < 13; j++)

{

if (Resistor\_color[j] == temp\_color) Resistor\_value[i] = j;

}

if (Resistor\_value[i] == -1)

{

printf("잘못 입력하였습니다.\n");

i--;

}

}

printf("마지막 오차 칼라를 입력하세요. ", i + 1);

scanf("%\*c%c", &temp\_color); // \n을 지우는 방법

for (int j = 0; j < 13; j++)

{

if (Resistor\_color[j] == temp\_color) Resistor\_value[i] = j; // 잘 못 입력 했을 때의 처리 부분을 추가하세요.

}

resistor\_value = 0;

/\*

for (i = 0 ; i < no\_band; i++)

{

printf("%d %d \n", i, Resistor\_value[i]);

}

\*/

if (no\_band == 4)

{

//

resistor\_value = (Resistor\_value[0] \* 10 + Resistor\_value[1]) \* (int)pow(10.0, (double)Resistor\_value[2]);

if (Resistor\_value[3] == 12) tolerance = 20;

else tolerance = (Resistor\_value[3] - 9) \* 5;

}

if (no\_band == 5)

{

//

resistor\_value = (Resistor\_value[0] \* 100 + Resistor\_value[1] \* 10 + +Resistor\_value[2]) \* (int)pow(10.0, (double)Resistor\_value[3]);

if (Resistor\_value[4] == 12) tolerance = 20;

else tolerance = (Resistor\_value[4] - 9) \* 5;

}

printf("저항값은 = %d 입니다.\n", resistor\_value);

printf("오차값은 = %3d %%입니다.\n\n", tolerance);

}

* 다른 점

1. 전역변수 선언으로 int, char 배열을 switch case 문 대신 사용
2. Do – while 반복문을 사용하여 밴드의 수가 4, 5가 아니면 반복하게 만들었음
3. 오차값을 일반 %로 표현해 간단하게 표현함
4. 색을 한번에 입력 하는 것이 아닌 오차값과 칼라값을 따로 구분해서 입력하게 함
5. 오차값을 스위치 문을 사용하는 대신 if-else문과 숫자값을 이용해 계산되게 함

* 개선해야 할 점.

1. Switch case 문대신 배열을 이용해서 코드 간소화
2. 틀렸을 상황을 생각해 종료 대신 반복하게 만들기
3. 숙제 코드에 char 전역변수에서 g가 중복되어서 실행은 가능하나 오류가 생깁니다.
4. 숙제 코드에서 int 배열에 -1 한 개만 선언 되어있어서 밴드 값 첫번째에선 잘 되지만 두번째에서 잘못 입력 시 코드가 그대로 입력되기 때문에 {-1,-1,-1,-1, }로 선언하여 반복되게 변경해야 합니다. int Resistor\_value[5] = { -1, }; >> int Resistor\_value[5] = { -1, -1, -1, -1, };
5. 오차 칼라를 잘 못 입력 했을 경우 코드

do {

printf("마지막 오차 칼라를 입력하세요. ", i + 1);

scanf("%\*c%c", &temp\_color); // \n을 지우는 방법

} while (temp\_color!='S' && temp\_color!='N' && temp\_color!='G');

Do while 문을 사용하여 입력 해 보았습니다.S, N, G 가 아닐 경우 반복하게 함

텍스트이(가) 표시된 사진

자동 생성된 설명