#include <stdio.h>

#include <math.h>

int select;

long double value = 0;

int findcolor();

char color[5] = { 0, 0, 0, 0, 0 };

void ohm();

int main(void) {

printf("4색코드는 4, 5색코드는 5를 입력해주세요: ");

scanf("%d", &select);

예시에서는 do while문으로 올바른 값(4, 5)을 입력하였는지의 체크를 하는지에대한 유무가 다르고

올바른 값을 입력하였는지 체크가 필요합니다

if (select == 4) {

findcolor();

value = color[0] \* 10 + color[1];

value \*= pow(10, color[2]);

ohm();

return 0;

}

else if (select == 5) {

findcolor();

value = color[0] \* 100 + color[1] \* 10 + color[2];

value \*= pow(10, color[3]);

ohm();

return 0;

}

}

예시에서는 입력받은 값을 미리 입력된 배열과 비교하여 같을시

배열에 입력하는 방법이었고

저는 함수에 if문을 여러 개 사용하고 비교해서 같을시 배열에 넣는 방식이었습니다 그리고 이 방법을 사용하면서 함수를 여러 번 호출하게 되고 많은 수의 if문으로 인해 코드가 길어지게 됩니다.

예시에서 한 방법처럼 배열을 사용하여 미리 입력된 값과 비교하는 것이 훨씬 간결하고 코드가 짧아지는점에서 개선해야한다고 생각합니다

또한 예시 처럼 제곱을 위 식에 괄호를 치고 곱해서

한줄로 통합하는 것이 더 좋아보입니다

int findcolor() {

for (int i = 0; i < select; i++) {

printf("색깔을 입력하세요 (검=B,갈=b,빨=r,주=o,노=y,녹=g,청=u,보=v,회=s,흰=w,골드=G,은=S,없음=N): ");

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

if (color[i] == 'B')

color[i] = 0;

else if (color[i] == 'b')

color[i] = 1;

else if (color[i] == 'r')

color[i] = 2;

else if (color[i] == 'o')

color[i] = 3;

else if (color[i] == 'y')

color[i] = 4;

else if (color[i] == 'g')

color[i] = 5;

else if (color[i] == 'u')

color[i] = 6;

else if (color[i] == 'v')

color[i] = 7;

else if (color[i] == 's')

color[i] = 8;

else if (color[i] == 'w')

color[i] = 9;

}

}

위에서 말씀 드렸던거처럼 배열을 이용한 비교 시 불필요한 함수가 됩니다 개선을 한다면 잘못된 값을 입력하였을 때 체크하는 부분을 만들어야 할거같습니다

void ohm() {

if (select == 4) {

if (color[3] == 'G')

printf("저항은 %lf ~ %lf입니다.", value \* 0.95, value \* 1.05);

else if (color[3] == 'S')

printf("저항은 %lf ~ %lf입니다.", value \* 0.9, value \* 1.1);

else if (color[3] == 'N')

printf("저항은 %lf ~ %lf입니다.", value \* 0.8, value \* 1.2);

}

else if (select == 5) {

if (color[4] == 'G')

printf("저항은 %lf ~ %lf입니다.", value \* 0.95, value \* 1.05);

else if (color[4] == 'S')

printf("저항은 %lf ~ %lf입니다.", value \* 0.9, value \* 1.1);

else if (color[4] == 'N')

printf("저항은 %lf ~ %lf입니다.", value \* 0.8, value \* 1.2);

}

}

예시에서는 입력된 저항값에 9를 뺀후 5를 곱해서

오차의 퍼센티지를 출력해주었습니다

저는 계산된 저항값에 마지막 컬러값만큼의 저항을 곱해서

저항의 최대 최소 범위를 구하였습니다

이 또한 코드가 길어지게 되는 불필요한 함수라고 생각하고 배열에

통합하여 간단한 수식을 통해서 퍼센트로 나타내는 것이

훨씬 효율적인 방법이 될거같습니다

추가로 예시 코드의 Resister\_value의 값에 -1을 모두 채워넣지 않아서

두번째부터 잘못된 코드값을 입력할 시에 정상적으로 걸러내지 못합니다

-1값을 모두 채워넣어야 할 것 같습니다

그리고 Resistor\_color 배열에 모든 색상이 들어가있어서

마지막 컬러가 아닌것을 금색, 은색, 없음으로 해도 문제가 발생하지 않는 부분이있습니다

따라서 resistor\_color를 앞 10개의 색상과 뒤 3개의 색상

두개의 배열로 나누고 체크부분 for문의 j값을 9까지로 수정하고

마지막 오차 컬러도 뒤 3개의 색상 배열을 이용하여

for문을 3개짜리의 배열로 체크하여 잘못된 값을 입력하였는지 여부를 확인할 수 있을 것 같습니다