```
1  #include<stdio.h>
2  #pragma warning(disable:4996)
3
4  #define CM_PER_IN 2.54
5  #define IN_PER_CM 1/2.54
6
7  void cm_to_in(void);
8  void in_to_cm(void);
9  /*---------------------------------------------------------------------*/
10 void main(void)
11 {
12   int iSelect;
13
14   printf("Enter 1 to convert inches to centimeters.\n");
15   printf("Enter 2 to convert centimeters to inches.\n");
16   printf("Enter 1 or 2 => ");
17   scanf ("%d",&iSelect);
18
19   if (iSelect == 1)
20     in_to_cm ();
21
22   if (iSelect == 2)
23     cm_to_in ();
24
25   if ((iSelect != 1) && (iSelect != 2))
26     printf("You did not enter a 1 or 2 \n");
27 }
28 /*---------------------------------------------------------------------*/
29 void in_to_cm(void)
30 {
31   float fInch, fCm;
32
33   printf("\nEnter value in inches =>  ");
34   scanf("%f",&fInch);
35   fCm = fInch * CM_PER_IN;
36   printf("There are %0.2f cms in %0.2f inches.\n",fCm, fInch);
37 }
38 /*---------------------------------------------------------------------*/
39 void cm_to_in(void)
40 {
41   float fInch, fCm;
42
43   printf("\nEnter value in cm =>  ");
44   scanf("%f",&fCm);
45   fInch = fCm * IN_PER_CM;
46   printf("There are %0.2f inches in %0.2f cms.\n",fInch,fCm);
47 }
48 /*
49 Enter 1 to convert inches to centimeters.
50 Enter 2 to convert centimeters to inches.
51 Enter 1 or 2 => 1
52
53 Enter value in inches =>  10
54 There are 25.40 cms in 10.00 inches.
55
56 Enter 1 to convert inches to centimeters.
57 Enter 2 to convert centimeters to inches.
58 Enter 1 or 2 => 2
59
60 Enter value in cm =>  5
61 There are 1.97 inches in 5.00 cms.
62 */
63
```

Commented [KW1]: # define constant float values.

Commented [KW2]: Function prototypres

Commented [KW3]: Save variable iSelect as an interger.

Commented [KW4]: Compare data stored in iSelect to number 1. If equal, execute user function in_to_cm().

Commented [KW5]: Curly braces not needed due to only having a single statement after the test condition.

Commented [KW6]: Compare data stored in iSelect to number 2. If equal, execute user function cm_to_in().

Commented [KW7]: Perform a Boolean test to see if iSelect contains numbers not equal to 1 and 2.

Commented [KW8]: Because three if() statements were used, lines 19, 22, and 25, all, some, or none of them may or may not be executed depending on the test conditions.

Commented [KW9]: Result of lines 14-17.

Commented [KW10]: Result of lines 29-37.

Commented [KW11]: Result of lines 14-17.

Commented [KW12]: Result of lines 39-47.

```
1  #include<stdio.h>
2  #include<conio.h>
3  #pragma warning(disable:4996)
4
5  #define  DOL_PER_HR  10.0
6  #define  OVR_PER_HR  (DOL_PER_HR * 1.5)
7
8  /*-------------------------------------------------------------------*/
9  void main (void)
10 {
11   float fPay, fHours;
12
13   printf("Enter the number of hours worked in one week. => ");
14   scanf ("%f",&fHours);
15
16   if (fHours <= 40.0)
17     {
18     fPay = fHours * DOL_PER_HR;
19     printf("\nYour pay for the week is $%0.2f. \n", fPay);
20     }
21   else
22     {
23     fPay = (fHours - 40.0) * OVR_PER_HR + 40.0 * DOL_PER_HR;
24     printf("\nYour pay for the week is $%0.2f." , fPay);
25     }
26 }
27 /*
28 Enter the number of hours worked in one week. => 45
29
30 Your pay for the week is $475.00.
31
32 Enter the number of hours worked in one week. => 30
33
34 Your pay for the week is $300.00.
35 */
36
```

Commented [KW13]: Because DOL_PER_HR was defined on line 5, it can now be used on line 6 to calculate a new value.

Commented [KW14]: Curly braces are needed since two statements are required to be executed after the test condition is true.

Commented [KW15]: This will always execute if the if is false. Note that there is no test condition for a else control structure.

Commented [KW16]: Result of the else being executed.

Commented [KW17]: Result of the if() being executed.

```
#include <stdio.h>
#pragma warning(disable:4996)

void checknum(int num);
//---------------------------------------------------------------------
void main(void)
{
            int number;

            printf("Enter a number from 0 to 19   ");
            scanf("%d", &number);

            checknum(number);
}
//---------------------------------------------------------------------
void checknum(int num)
{
            if ((num >= 0) && (num < 5))
                        puts("The number is 0, 1, 2, 3, or 4. \n");

            else if ((num >= 5) && (num < 10))
                        puts("The number is 5, 6, 7, 8, or 9. \n");

            else if ((num >= 10) && (num < 15))
                        puts ("The number is 10, 11, 12, 13, or 14. \n");

            else if ((num >= 15) && (num < 20))
                        puts("The number is 15, 16, 17, 18, or 19. \n");

            else
                        puts("The number is below 0 or above 19. \n");
}
/*
Enter a number from 0 to 19   2
The number is 0, 1, 2, 3, or 4.

Enter a number from 0 to 19   7
The number is 5, 6, 7, 8, or 9.

Enter a number from 0 to 19   14
The number is 10, 11, 12, 13, or 14.

Enter a number from 0 to 19   15
The number is 15, 16, 17, 18, or 19.

Enter a number from 0 to 19   20
The number is below 0 or above 19.

Enter a number from 0 to 19   -3
The number is below 0 or above 19.

*/
```

Commented [KW18]: if()..else if()..else structure implemented using Boolean test conditions to test for bounded numerical values.

Commented [KW19]: When testing program logic, you should test all possible Boolean conditions to ensure you have not made any logical errors.