**Homework 8**

1. Write a function IsPrime(…) that has a single parameter x of type integer. If x is a prime number, the function returns 1; otherwise, the function returns 0
2. Write a function EvalPoly(…)that expects four float parameters a, b, c, and x. The function should return the value ax2 + bx + c.
3. Write a function that rounds a number to a given number of decimal places(有效位數) and returns the rounded value as the function result.

x

Round of (x)

Round

(四捨五入)

places

(有效位數)

For example, the call Round (7.8257 , 2) would return the value 7.83.

Hint: To round a number to the nearest integer, add 0.5 and then truncate

the sum( discarding the fraction part 去掉小數部份)

Ex2:

Round (7.8257, 3)

7.826

782.57

+ .5

783.07

Ex1:

Round (7.8257, 2)

7.83

Hint:

Just as for the field width specifier, the programmer may use a number or an asterisk as a precision specifier.

The asterisk indicates that the actual value of the precision specifier will be one of the additional parameters to the printf call.

For example,

Case 1: printf ("%.2f\n", 3.675); will print: 3.68

Notice that printf rounds the number.

Case 2: Assume that width =3

Ex1: printf ("%.\*f\n", width, 10.4);

assuming that the current value of width is 3, will print: 10.400

Ex2: printf ("%\*d%\*d\n", width, 10, width, 12);

assuming that the width is 6, will print: 10 12

Ex3:

#include<stdio.h>

int main(){

int wid;

double data=1.987654321;

for(int i=0;i<5; i++)

{

printf("Please enter the width:");

scanf(" %d", &wid);

printf("data=%.\*lf\n",wid,data );

}

printf ("\n%\*d%\*d\n", wid, 12, wid, 456);

return 0;

}

1. Write a program that prints a calendar for a year. Prompt the user for the year and print the year and the calendar.

Hint1:

January 1 in year x begins on day:

 mod 7

(a).Where  denotes the greatest integer less than or equal to *x*.

(b). M mod n denotes the remainder when m is divided by n.

(c). Sunday corresponds to 0, Monday to 1, and so on.

For example, if *x* = 1998,

 mod 7

= (1998 +499- 19 +4) mod 7 = 4

Thus, January 1, 1998 begins on Thursday.

Hint2: Year *x* is a leap year if

*x* is divisible by 4 and not by 100

*or x* is divisible by 400

For example:

(a).1998 is divisible by neither 4 nor 400. So 1998 is not a leap year.

(b).1996 is divisible by 4 and not by 100. So 1996 is a leap year.

(c).2000 is divisible by 400. So 2000 is a leap year.

(d).1990 is divisible by 4 and by 100, and is not divisible by 400.

So 1990 is not a leap year.

1. A control system applies a force to an actuator (致動器) proportional to the voltage of a signal coming into the control system.

It is desired not to allow the actuator to quiver back and forth in the presence of small corrections near the zero-force point(若力太小,則忽略不計,以免actuator擺來擺去)

More force is required for the actuator to move to the left (negative direction of motion) than is required for motion to the right (positive direction motion).

Assume that the transfer function (the relationship between the voltage and the movement) of the actuator is

* Voltage less than –0.2 volt: Actuator moves 1 cm/volt in the negative direction.
* Absolute value of voltage less than or equal to 0.2 volt: No motion.
* Voltage great than 0.2 volt: Actuator moves 2 cm/volt in the positive

Direction.

Write a function force (…) to compute the total motion for any signal input

.

Write a main program that repeatedly calls the force ( …) function using

an input signal stream such as :

-10.0 v, –8.0 v, -0.21 v, -0.20 v, -0.05 v, 1.5 v, 0.00 v, 4.5 v,

10.0 v

The main program should also take as user input an initial position of the actuator and should output a final position resulting from applying the signals of the given control stream.

For one test, simulate the effect of the given voltages for an initial position of 1.5 cm to find the final position of the actuator.