Homework 7

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.
   Please write a main program that inputs one integer and pass this integer to the IsPrime(…) to test this function.
   Stop the program when inputting CTRL+Z.

   Input/Output Example:
   Input a number: 13
   13 is prime

   Input a number: 9
   9 is not prime

   Input a number: 2
   2 is prime

   Input a number: 1
   1 is not prime

   Input a number: ^Z

2. Write a function EvalPoly(…) that expects four float parameters a, b, c, and x. The function should return the value $ax^2 + bx + c$.
   Please write a main program that inputs these four numbers to test this EvalPoly(…) function.
   Please show the answer to the 5th decimal place.
   Stop the program when inputting CTRL+Z.

   Input/Output Example:

   Input a, b, c, x: 1.11 2.22 3.33 25.5
   ax^2+bx+c = 781.71750

   Input a, b, c, x: 3.33 2.22 1.11 25.5
   ax^2+bx+c = 2223.05250

   Input a, b, c, x: ^Z

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.

A. For example, the call Round (7.8257 , 2) would return the value 7.83. and calling Round (1.23432, 3) will return the value 1.234. Stop the program when inputting CTRL+Z.

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)

Ex1:

Round (7.8257, 2)

⟹ 7.83

782.57

\+ .5

783.07

Ex2:

Round (7.8257, 3)

⟹ 7.826

Input/Output Example:

```
Please input a number and a rounded place:10.22 0
The result is 10
Please input a number and a rounded place:5.67999 4
The result is 5.6800
Please input a number and a rounded place:3.14159 2
The result is 3.14
Please input a number and a rounded place:^Z
```

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;
}
```

4. Write a function printCal(…) that prints a calendar for a year. Prompt the user for the year and print the year and the calendar.
Stop the program when inputting CTRL+Z

Hint1:
January 1 in year x begins on day:

$$\left(x+\left\lfloor\frac{x-1}{4}\right\rfloor-\left\lfloor\frac{x-1}{100}\right\rfloor+\left\lfloor\frac{x-1}{400}\right\rfloor\right) \bmod 7$$

(a).Where $\lfloor x \rfloor$ 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,

$$\left(1998+\left\lfloor\frac{1998-1}{4}\right\rfloor-\left\lfloor\frac{1998-1}{100}\right\rfloor+\left\lfloor\frac{1998-1}{400}\right\rfloor\right) \bmod 7$$

= (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.

Input/Output Example:

Please input the year: 2021

January

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

February

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

March

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20

```
21  22  23  24  25  26  27
28  29  30  31
```

April

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
                1   2   3
 4   5   6   7   8   9  10
11  12  13  14  15  16  17
18  19  20  21  22  23  24
25  26  27  28  29  30
```

May

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
                        1
 2   3   4   5   6   7   8
 9  10  11  12  13  14  15
16  17  18  19  20  21  22
23  24  25  26  27  28  29
30  31
```

June

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
            1   2   3   4   5
 6   7   8   9  10  11  12
13  14  15  16  17  18  19
20  21  22  23  24  25  26
27  28  29  30
```

July

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
                1   2   3
 4   5   6   7   8   9  10
11  12  13  14  15  16  17
18  19  20  21  22  23  24
25  26  27  28  29  30  31
```

August

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
 1   2   3   4   5   6   7
 8   9  10  11  12  13  14
15  16  17  18  19  20  21
22  23  24  25  26  27  28
29  30  31
```

Septemper

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
                1   2   3   4
 5   6   7   8   9  10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30
```

October

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
                        1   2
 3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27  28  29  30
31
```

November

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
    1   2   3   4   5   6
 7   8   9  10  11  12  13
14  15  16  17  18  19  20
21  22  23  24  25  26  27
28  29  30
```

December

```
-----------------------------------
Sun Mon Tue Wed Thu Fri Sat
-----------------------------------
                1   2   3   4
 5   6   7   8   9  10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30  31
============================
```

Please input the year: ^Z

5. 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.
Please show the answer to the 4th decimal place.
Please follow the example to let user continuously input the initial position and **signal stream** and stop when inputting CTRL-D and **CTRL-Z**, respectively.

Input/Output Example:

Please input initial position: 1.5
Voltage:
-10.0 -8.0 -0.21 -0.20 -0.05 1.5 0.00 4.5 10.0
^Z
Final position: 15.2900

Please input initial position: ^D