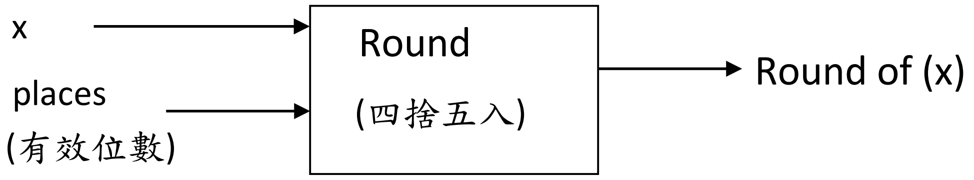
Lab08

1. Homework Problem I

a) 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

b) Write a function EvalPoly(…)that expects four float parameters a, b, c, and x. The function should return the value $ax^2 + bx + c$.

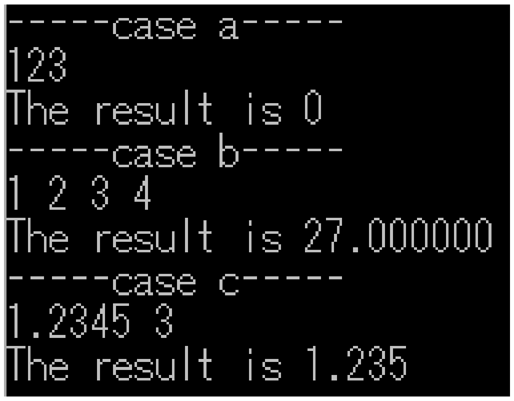
c) Write a function that rounds a number to a given number of decimal places(有效位數) and returns the rounded value as the function result.



For each of 3 cases, get input from the user and call each function. Then print out the results.

Print the result of case c by the method mentioned in homework.

Sample Input/Output:



```
-----case a-----
123
The result is 0
-----case b-----
1 2 3 4
The result is 27.000000
-----case c-----
1.2345 3
The result is 1.235
```

2. Homework Problem II

A control system applies a force to an actuator 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.

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.

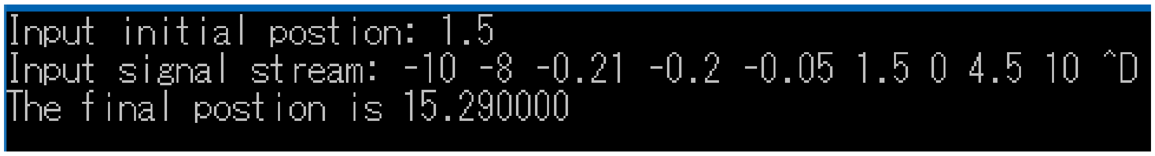
Write a main program that **repeatedly calls the force (…)** function using an input signal stream such as :

-10 -8 -0.21 -0.2 -0.05 1.5 0 4.5 10

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.

<u>The input stream ends when the input is Ctrl+Z or Ctrl+D.</u>

Sample Input/Output:



```
Input initial postion: 1.5
Input signal stream: -10 -8 -0.21 -0.2 -0.05 1.5 0 4.5 10 ^D
The final postion is 15.290000
```

3. What day is today?

Please define the following functions

- **int Date2Day(int y, int m, int d)**
  Don't use switch-case. & Don't use if-else over 2th order.
  Get the day of the date. And we define that "0" represents "Sunday", "1" represents "Monday", and so on. For example, 2019/11/25 -> 1 (Monday).
- **void printDay(int day)**

  Print out the mapping of the outputs from Date2Day to the Monday, Tuesday, ……. You may use switch-case to implement it

- **void printCalendar(int y, int m, int d)**

  Print out the monthly calendar of the month. You may use Date2Day to get the day of the first day of the month. Please mark the corresponding days as specified text colors and background colors, which is shown as below sample output.

Hint: If the year is divisible by 4 and not by 100, or the year is divisible by 400, it is a leap year.

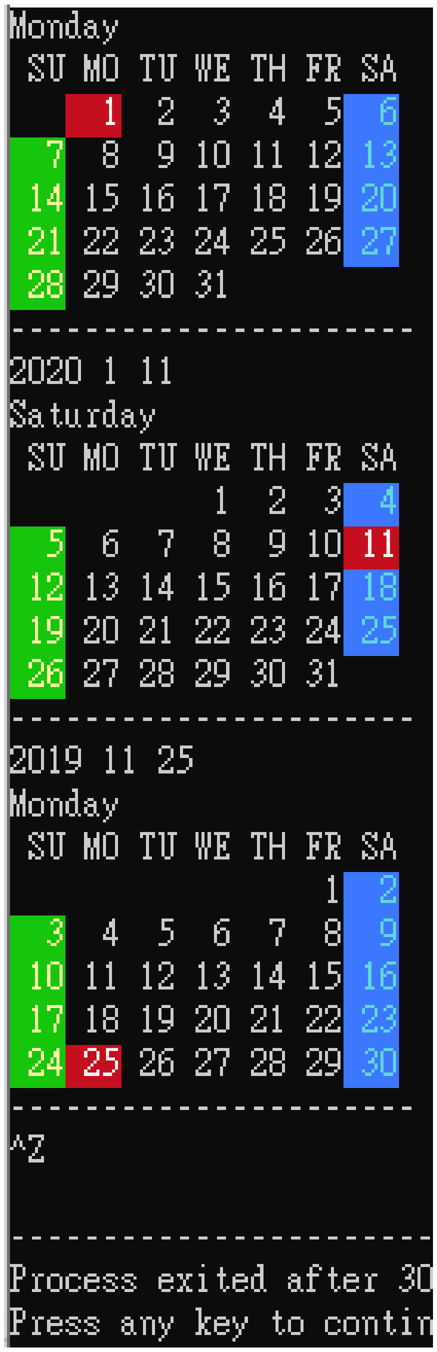
Hint: The day of the first day of a year can be calculated with following formula.

$$Day = \left(\mathrm{y} + \left\lfloor\frac{y-1}{4}\right\rfloor - \left\lfloor\frac{y-1}{100}\right\rfloor + \left\lfloor\frac{y-1}{400}\right\rfloor\right) mod\ 7$$

The main program should prompt user to input a **date** and print out the corresponding **day** as well as the **calendar** of the month of the date.

The program stops when the input is Ctrl+Z or Ctrl+D.

Sample Input/Output:



```
Monday
 SU MO TU WE TH FR SA
        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
----------------------
2020 1 11
Saturday
 SU MO TU WE TH FR SA
           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
----------------------
2019 11 25
Monday
 SU MO TU WE TH FR SA
                 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
----------------------
^Z


-----------------------
Process exited after 30
Press any key to contin
```

4. Power Modulo Problem

Define the function long long powmod(long long n, long long k, long long m) to compute n^k % m. For k is very big, you can't multiple number n with k times to complete it.

As the result, you may use the algorithm showed as below,

a = b * c -> a % k = ((b % k) * (c % k)) % k

That is,

a^2 % k = ((a % k) * (a % k)) % k

a^4 % k = ((a^2 % k)*(a^2 % k)) % k

a^8 % k =……

For example,

$111_{dec} = 1101111_{bin}$

13^{111} % 113 = (13^{64} * 13^{32} * 13^8 * 13^4 * 13^2 * 13^1) % 113

= ((13^{64} %113)*(13^{32} %113)*(13^8 %113)*(13^4 %113)* (13^2 %113)

*(13^1 %113)) % 113

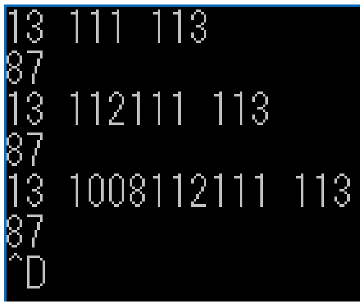
Every multiplication must follow by modulo to avoid overflow.

For each test case, enter three integer n, k and m as inputs, and output the result. The program stops when the input is Ctrl+Z or Ctrl+D.

Sample Input/Output:

13 112111 113 -> 87

13 1008112111 113 -> 87



```
13 111 113
87
13 112111 113
87
13 1008112111 113
87
^D
```