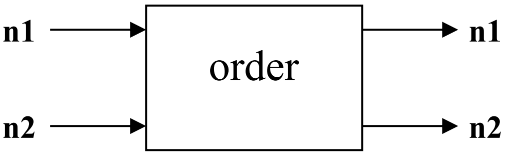
Homework 8

No global variable or array allowed in this Homework.

1. Write a subroutine order: void order(…)

double n1, n2 : **call by address**



n1, n2 : call by address

After the calling and execution of order:

The smaller of its two argument values is stored in its first actual parameter and the larger is stored in its second actual parameter. Please do not print any message in the function. Let the user continuously input n1 and n2 until inputting **Ctrl+Z**. **Print the n1 and n2 after ordering in the main function**.

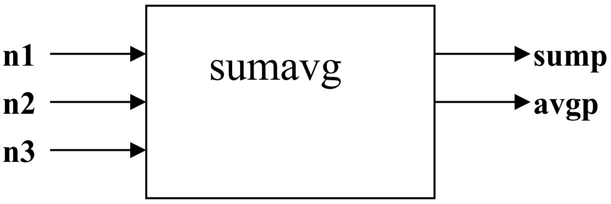
Input/Output Example:

```
input n1, n2: 3.56 1.02

--- Before ordering ---
n1 = 3.560000, n2 = 1.020000
--- After ordering ---
n1 = 1.020000, n2 = 3.560000

input n1, n2: ^Z
```

2. Write a subroutine sumavg that has three type double input parameters n1, n2, n3 and two output parameters sump, avgp.



(Actually subroutine sumavg has five parameters:
n1, n2, n3: passed by value
sump, avgp: passed by address)

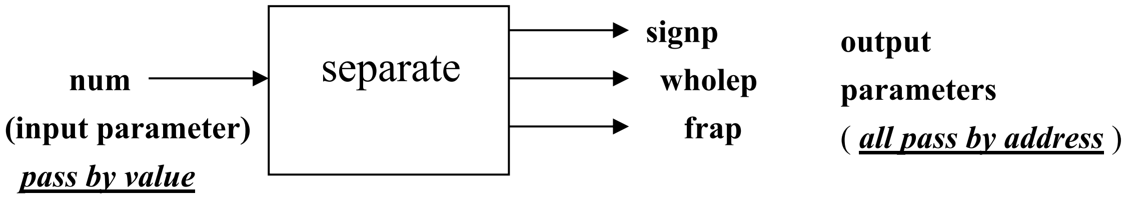
The subroutine computes the sum and average of its three input arguments. Please do not print any message in the function. Let user continuously input n1, n2, n3 until Ctrl+Z

Input/Output Example:

```
input n1, n2, n3: 1.23 3.45 5.67
sum = 10.350000
avg = 3.450000

input n1, n2, n3: ^Z
```

3. **Write a subroutine separate:**



Function separate finds the sign, whole number magnitude, and fractional parts of its input parameter num. (可使用 math.h 之 function)

<u>Prototype for separate:</u>

```
 void separate( double num,      // input value to be split
                char * signp,    // output -- sign of num
                int * wholep      // output – whole number magnitude of num
                double * fracp) // output – fractional part of num
```

Let user continuously input num until Ctrl+D. Please do not print any message in the function. Print the signp, wholep, and fracp in the main function.

Input/Output Example:

```
Input num: 1.2345
1.234500 = (+) ( 1 + 0.234500 )
Input num: 99.9999
99.999900 = (+) ( 99 + 0.999900 )
Input num: 0
0.000000 = (+) ( 0 + 0.000000 )
Input num: ^D
```

4. **Write a program for an Automatic Teller Machine that dispenses money.(發錢功能)**
   The user should enter the amount desired(a multiple of 10 dollars, 10 元的倍數) and the machine dispenses this amount using <u>the least number of bills</u>.
   The bills dispensed are 50s, 20s, and 10s. Write a function that determines how many of each kind of bill to dispense.

   <u>Prototype for ATM:</u>

   bool ATM(int dollars, int* n50, int* n20, **int*** n10);

   - dollars is the desired amount inputting from the user. **(Be sure to handle the**

exception when the amount is not a multiple of 10 dollars.)

- In ATM function, it will produce the numbers of bills 50, 20, and 10 respectively dispensed by Automatic Teller Machine and **pass these three data** back to the main by using “**call by address”**.
- Please do not print any message in the function ATM.

Let the user continuously input dollars until inputting Ctrl+D.
Print the least number of bills in the main function.

Input/Output Example:

```
Input amount: -120
Illegal input!!

Input amount: 321
Illegal input!!

Input amount: 0
50s: 0
20s: 0
10s: 0

Input amount: 89100
50s: 1782
20s: 0
10s: 0

Input amount: 930
50s: 18
20s: 1
10s: 1

Input amount: ^D
```

5. **Write a function that translates the amount of Arabic number into Chinese representation. Let user continuously input until Ctrl+Z.**

EX1: 7609802 ------ 柒佰陸拾萬玖仟捌佰零貳元整
EX2: 1009040300 -----拾億零玖百零肆萬零三佰元整

Input/Output Example:

```
Input: 123
Output:  壹百貳拾參元整
```

Input: 1234567890
Output: 十貳億參千肆百伍拾陸萬柒千捌百玖拾元整

Input: 1009040300
Output: 十億零玖百零肆萬零參百元整