Lab 11/22

Please do not use array, global variable and recursive function in this lab.

Lab Part

1. Encoder
   a. Right shift the code x.
      Right shift the code x N times, and the overflow digit will be put to the front.
      Example: 12345 right shifts 678 digits → 34512
   b. Add the bias
      Add the bias to the code x.
      Example: 34512+90123 → 124635
   c. Truncate the extra digits.
      If the length of result is longer than original length, the extra digits will be truncated.
      Example: 124635→ 24635

   (1) Please write 5 functions with prototype:
      int length(…) → calculate the string length of x
      void rightShift(…) → step a
      void addBias(…) → step b
      void truncate(…) → step c
      int encode(…) → implement of the encoding algorithm above.
      All of these functions should accept a parameter x (pass by value or address), you can add other parameters if needed.

   (2) The only self-defined function could be invoked in "main" is "encode".
   (3) Let user continuously input x, N, bias until ctrl+Z.
   (4) The computation time should be only depend on the length of x,
      (e.g. "123 999999 456" should be execute as fast as "123 9 456")

Input/Output Example:

```
input x, N, bias: 12345 678 90123
24635

input x, N, bias: 123 456 789
912

input x, N, bias: ^Z
```

2. Vector Operations

Please design following functions that works on two 3D vectors:

Op	Functions	No. parameters	Description
+	add	9	v1 + v2
-	subtract	9	v1 - v2
*	dot	7	v1 * v2
x	cross	9	v1 x v2
	show	3	print “(x, y, z)“

In addition, write a function “void compute(…)” retrieve two vectors and an operator as its arguments.

(1) Pass the vector by value and retrieve the result by passing address.

(2) Let user continuously input x1, y1, z1, op, x2, y2, z2 until ctrl+Z.

(3) Show the answer to the 3nd decimal place.

(4) The “show” function should only be invoked in the “compute” function.

Ref: https://en.wikipedia.org/wiki/Cross_product

Input/Output Example:

```
Input: 1.2 3.4 5.6 + 7.8 9.0 1.2
(9.000, 12.400, 6.800)

Input: 1.2 3.4 5.6 - 7.8 9.0 1.2
(-6.600, -5.600, 4.400)

Input: 1.2 3.4 5.6 * 7.8 9.0 1.2
46.680

Input: 1.2 3.4 5.6 x 7.8 9.0 1.2
(-46.320, 42.240, -15.720)

Input: 1.2 3.4 5.6 ? 7.8 9.0 1.2
Invalid operator

Input: ^Z
```

3. Please write three function to calculate the roots of following equation:

$$a * sin^2(x) + b * sin(x) + c = 0$$

Let $y = \sin(x)$, then equation become

$$ay^2 + by + c = 0$$

Then we can find its roots by Bhaskar's equation

$$y = \left\{ \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right\}$$

Finally, take arcsine on y would get the actual roots of equation

$$\arcsin(y) = y + \sum_{n=1}^{N} \underbrace{\frac{1 * 3 * \ldots * (2n-1)}{2 * 4 * \ldots * (2n)} \frac{y^{2n+1}}{2n+1}}_{a_n} \quad , |y| \leq 1$$

Please definite and use follow three function to calculate the ***x*** (roots)

(1)***double sqrt_f(…);***

=> pass one parameter **(*x*)** called by value and return its square roots **(r)**. Please use Newton's method and end with $|r^2 - x| < 10^{-10}$, return ***r***.

(2)***bool arcsin_f(…);*** => pass one parameter **(y)** called by address, return that function argument **(y)** is in the correct range $|y| \leq 1$ or not, then store the result of equation above which ending with $|a_n| < 10^{-13}$ if **(y)** is available .

(3)***int compute_root(…,…,…);*** => two parameters (***a, b***) called by address and one parameter (***c***) call by value.

⇨ **you will call function (1) and (2) in this function**

⇨ using a, b to stores (x_1, x_2) if there have real number solutions.

⇨ return a flag which is

0 if there is no real number root, 1 if there are two roots,

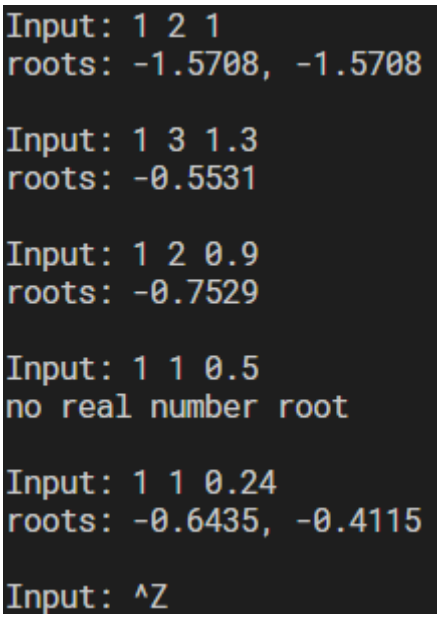
2 if the only root is stored in b, 3 if the only root is stored in a.

(4)Please print the result to 4th decimal place

(5)Let user continuously input (a, b, c) until entering Ctrl+Z.

(6)No math.h library.

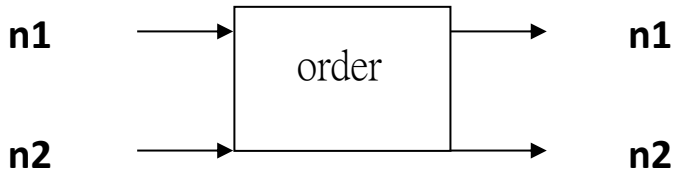
Input/Output Example:



Homework Part

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

double 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**

Output to the 7th decimal places for floating points.

Modification:

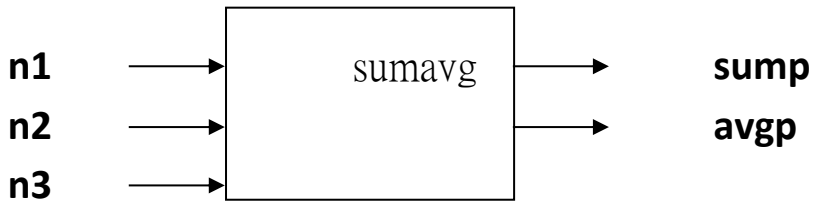
(1)**double** n1, n2 : **call by address** -> **double** n1, n2, n3 : **call by address**

(2)result: n1 > n2 > n3

Input/Output Example:

```
input n1, n2, n3: 1.23 4.56 7.89

--- Before ordering ---
n1 = 1.2300000, n2 = 4.5600000, n3 = 7.8900000
--- After ordering ---
n1 = 7.8900000, n2 = 4.5600000, n3 = 1.2300000

input n1, n2, n3: 7.89 4.56 1.23

--- Before ordering ---
n1 = 7.8900000, n2 = 4.5600000, n3 = 1.2300000
--- After ordering ---
n1 = 7.8900000, n2 = 4.5600000, n3 = 1.2300000

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

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



(Actually subroutine sumavg has <u>five parameters:</u>

n1, n2, n3: <u>passed by value</u>

sump, avgp: <u>passed by address</u>)

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

Modification:

(1)Write a subroutine sumavgpro with six parameters:

n1, n2, n3: passed by value

sump, avgp, prop: passed by address

(2)The subroutine computes the sum, average and product of the three numbers.

Input/Output Example:

```
input n1, n2, n3: 1.23 3.45 5.67
sum = 10.350000
avg = 3.450000
pro =  24.060645

input n1, n2, n3: 1.47 2.58 3.69
sum = 7.740000
avg = 2.580000
pro =  13.994694

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

6. 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 the least number of bills. The bills dispensed are 50s, 20s, and 10s. Write a function that determines how many of each kind of bill to dispense.

   Prototype for ATM:

   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.

Modification:
(1)50, 20, 10 -> 100, 50, 20, 10
(2)Modify the prototype for ATM

Input/Output Example:

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

Input amount: 321
Illegal input!!

Input amount: 0
100s: 0
50s: 0
20s: 0
10s: 0

Input amount: 98100
100s: 981
50s: 0
20s: 0
10s: 0

Input amount: 930
100s: 9
50s: 0
20s: 1
10s: 1

Input amount: ^D
```