Purpose: This program will utilize the MIPS assembly language. It will entail writing a series of functions and include the use of floating point instructions.

Program Specifications

A template file is included with data declarations and main already completed. You will need to fill in the four functions called by main. Do not make any changes to main or add data declarations.

Function 1 – Estimate Integer Square Root

Utilize the Newton method to calculate the integer square root of a number which will be passed an argument.

$$Estimate_{new} = (Estimate_{old}+Number/Estimate_{old})/2$$

Continue updating the estimation until $Estimate_{new}$ and $Estimate_{old}$ are within 1 of each other.

Return the final estimate.

Function 2 – Estimate Float Square Root

Use the Newton method as outlined above, but with floating point arguments and instructions. This time the function will continue to generate new estimates until it is within a specified tolerance of the previous estimate. Both the number and the tolerance level will be arguments to the function.

$$Estimate_{new} - Estimate_{Old} \geq -Tolerance \text{ and}$$
$$Estimate_{new} - Estimate_{Old} \leq Tolerance$$

Return the final estimate.

Function 3 – Print Integer Array

This function will output the elements of an integer array to the console. It must call the estimateIntegerSquareRoot function to determine how many elements will be printed on each line. The array to print and the number of elements will be passed as arguments. There is no return value.

Function 4 – Comb Sort (Ascending)
This function will sort an array of integer values in ascending order using the Comb Sort algorithm.

$$Gapsize_{new} = 10 \times gapsize_{old} / 13$$

Compare elements that are gapsize apart and swap if they are in the wrong order. After each iteration through the array, adjust the gapsize (minimum 1). If the gapsize is 1 and no swaps are performed in a single pass of the array, the algorithm stops.

The array to sort and the number of elements in the array will be passed as arguments. There is no return value.

Submission
Once you are satisfied with the program, upload the assembly source code (.asm) file to the class website.

Example Execution

```
The square root of 1742 is 41.
The square root of 4566 is 67.
The square root of 15135.0 is 123.02438354.
The square root of 911560.50 is 954.75677490.

Print Array Test:
1 1 1 1 1 1
1 0 0 0 0 1
1 0 0 0 0 1
1 0 0 0 0 1
1 0 0 0 0 1
1 1 1 1 1 1
1 1

Unsorted List:
377 148 641 -486 828 456 192 -742 -658 -139
801 -946 325 916 982 902 -809 858 -510 -713
-309 515 587 320 994 528 -617 -515 -123 294
644 -339 842 -441 -557 58 773 694 78 -744
-350 -424 -514 -679 402 -924 -178 315 509 173
44 -80 -340 905 -840 -210 671 -755 -809 731
-936 -414 627 -565 -749 -804 -456 -236 933 961
-675 -9 653 581 -567 916 738 343 684 -184
-789 -400 -941 145 933 230 -236 880 646 -926
982 221 -451 -783 331 -157 193 940 -818 270

Sorted List (Ascending):
-946 -941 -936 -926 -924 -840 -818 -809 -809 -804
-789 -783 -755 -749 -744 -742 -713 -679 -675 -658
-617 -567 -565 -557 -515 -514 -510 -486 -456 -451
-441 -424 -414 -400 -350 -340 -339 -309 -236 -236
-210 -184 -178 -157 -139 -123 -80 -9 44 58
78 145 148 173 192 193 221 230 270 294
315 320 325 331 343 377 402 456 509 515
528 581 587 627 641 644 646 653 671 684
694 731 738 773 801 828 842 858 880 902
905 916 916 933 933 940 961 982 982 994
```