ASSIGNMENT 4

Diwakar Prajapati (2018CS10330)

AIM: To compare time of execution of queries by plotting graph of Arbitrary Precision (Assignment 3) and GCC Compiler using GNUPLOT.

DESIGN DETAILS-
In all the function I have used char array to do operations, and have used char* to return the result from the function. It is due to this reason that whenever I have to executed a sequence of instructions I need to make a copy of data by creating char array from the pointer which takes extra time in executing instructions from my Arbitrary Precision.

ADDITION - I have first aligned the decimal point, thus filling extra zero at the front of back to the required operands, and then adding from right to left digit wise.

In am also handling in case if we are adding a negative number to a positive number by calling subtract.

Initially, carry = 0, sum=0;
sum = (carray + operand1[i] +operand2[i]) % 10
carry = (carray + operand1[i] +operand2[i]) / 10

The moment it finds decimal, it places decimal at the same position in result. Finally trimming the extra zero from the result.

SUBTRACTION - The algorithm is almost same as that of the addition, except, here I am also checking which operand is bigger and so subtraction accordingly. I am also handling the case where a negative number is being subtracted. Rest is same as addition

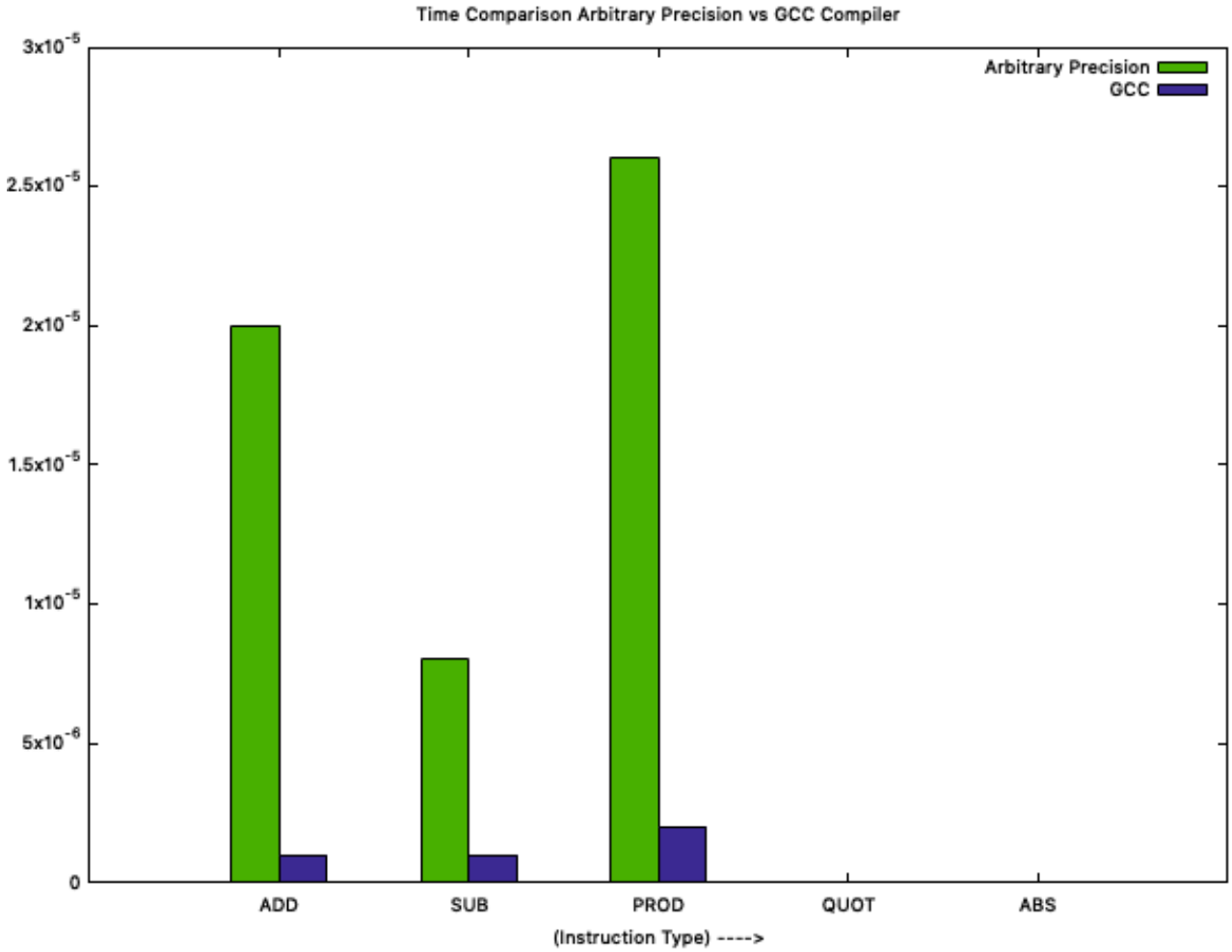
MULTIPLICATION - The algorithm used is the basic algorithm used by us in long multiplication:

```
   1234
 x  325
   6170
  2468x
 3702xx
 401050
```

I used this by multiplying the left bit of multiplier to multiplicand and storing the in result and then shifting and doing the same.

DIVISION - The algorithm used is Long Division Method. To get 20 decimal precision I have multiplied the dividend by 10^20(i.e. added 20 zeroes at the end) and then calculation the quotient and the balancing the decimal 20 places to the right that original answer.

SQUARE ROOT - I have used Babylonian method to find square root of a number. The is an algorithm easier to implement than the way we originally calculate square root manually using paper pen. But the former has higher time complexity than the latter ones.
GNUPLOT - I have created two arrays to store GCC time and time take by arbitrary precision. Used these array to create data.txt file where:
 -> 1st row is ADD <Avg_ArbPrecision_Time> <Avg_GCC_Time>
 -> 2nd row is SUB <Avg_ArbPrecision_Time> <Avg_GCC_Time>
 -> 3rd row is PROD <Avg_ArbPrecision_Time> <Avg_GCC_Time>
 -> 4th row is QUOT <Avg_ArbPrecision_Time> <Avg_GCC_Time>
 -> 5rth row is ABS <Avg_ArbPrecision_Time> <Avg_GCC_Time>
Now I have used this data.txt file to create plot bar graph by creating pipe to GNUPLOT. Same graph is given on next page.



RUN - Run the Makefile in Top directory, this command compiles all the C files, and creates the library files from the object files, creates executable file, links the library to the executable file.
I also maintain the project by shifting the object files and executable files to proper directories.

The main executable file “mainfile” is being located in exe folder

To run the executable file:

./exe/makefile input.txt output.txt Graph.png