**Summing Two Numbers from a File in Linux**

**Source Code:**

#include <time.h>

#include <sys/file.h>

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

int main()

{

double x, y, z;

int i, j = 0;

FILE \*ifp;

int data[10100];

int sum = 0;

int count = 0;

// clock function

x = clock(); /\*Strcat the clock\*/

ifp = fopen("numbers.dat", "r");

if (!ifp)

{

printf("\n\tcannot open input file\n\n");

return (1);

}

else

{

int size = sizeof(data) / sizeof(int);

for (i = 1; i <= size; i++)

{

fscanf(ifp, "%d", &data[i]);

//nested loop

for (j = i + 1; j <= size; j++)

{

fscanf(ifp, "%d", &data[j]);

if (i < j)

{

sum = data[i] + data[j];

if (sum == 100)

{

printf("\nSum is: %d + %d = %d\n", data[i], data[j], sum);

}

}

}

}

fclose(ifp);

}

y = clock();

z = (y - x) / CLOCKS\_PER\_SEC;

printf("\nThe time it took is: %lf seconds \n\n", z);

}

How to compile:

gcc main.c -o main

If the server show c99 mode , this is for “for loop”

gcc -std=c99 -o outputfile sourcefile.c

This will generate the executable main and you can run it using

./main

The data structure I have used in the program is array of long int type, the limitation on input is the file size cannot exceed more than 100,000 as it is the size defined in the program. The output has no certain limitations as it shows when the sum of two numbers match the pre determined random values generated within 10000.

Upon starting the program initializes the variables including the clock and opens the file in the read mode and check if it is opened or not. After that the file descriptor is passed to the readFile function, the declaration is as follows.

fscanf(ifp, "%d", &data[i]);

First reads the inputs file line by line and store the numbers on each line in the data structure and upon the end it returns the data structure where it is stored in a pointer declared in the main function.

This array of type long int is then passed to the twoSum function , the definition is as follows.

if (i < j)

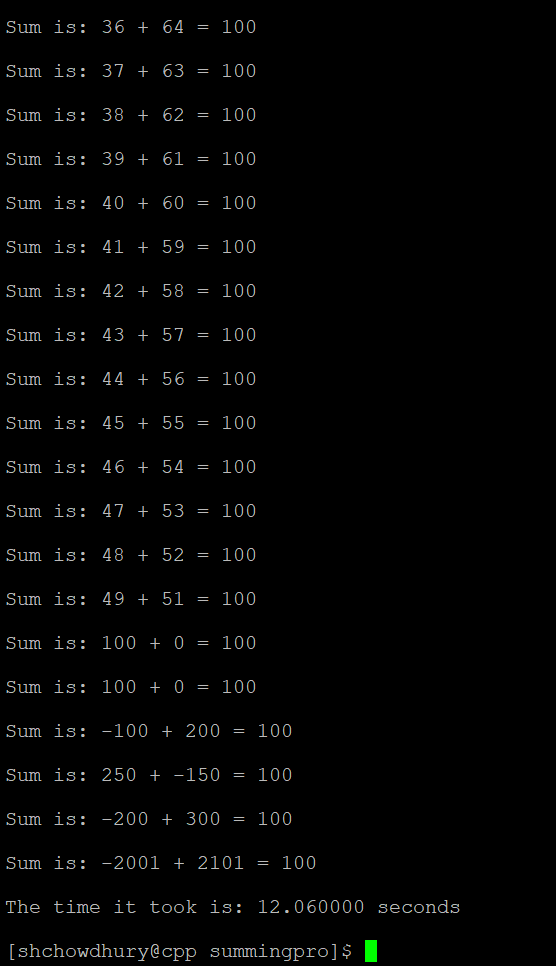
{sum = data[i] + data[j];

if (sum == 100)

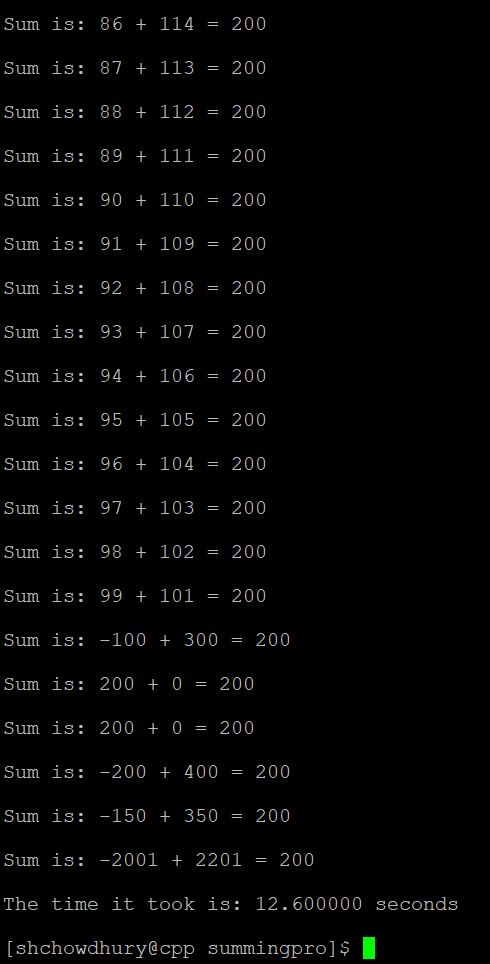
{printf("\nSum is: %d + %d = %d\n", data[i], data[j], sum);}

This function has a nested loop which has a complexity of O(n^2) and it sums the first number with the second, then the first with the third, the first with the forth, and so on. Then sum the second with the third, the second with the forth, and so on and it also generates a random numbers and if the sum of two numbers meets the random number generated then it prints a message.

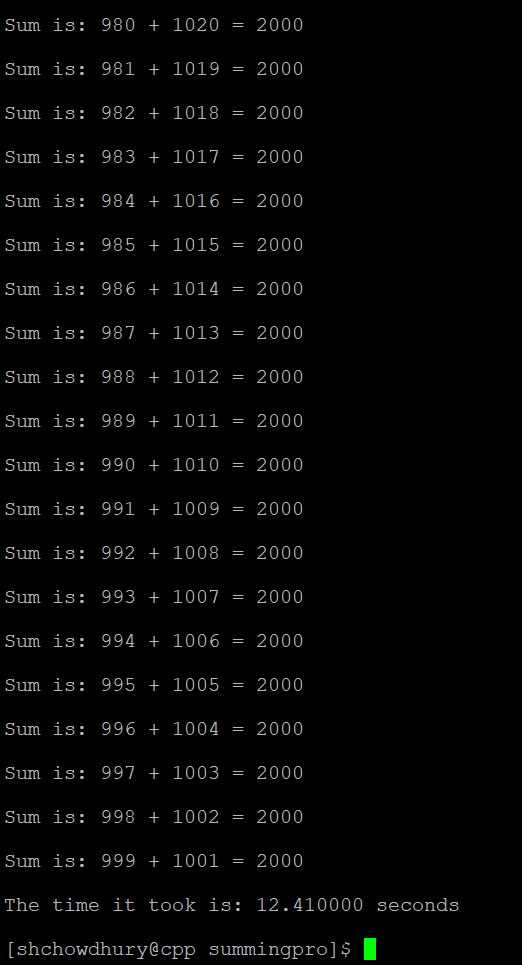
if (sum == 100)



if (sum == 200)



if (sum == 2000)



Conclusion:

When file size is decreasing the numbers of hits are decreasing which is resulting in the time taking to calculate, this result can vary and depends on the pre determined numbers and on the processing power of computer/server.