**Design and Analysis of Algorithm Lab**

**Lab 2 – Selvakumar G (22MAI1004)**

**Program : Execution Time Analysis: Insertion Sort**

Code:

#include <math.h>

#include <stdio.h>

#include<ctime>

#include<cstdlib>

#include <fstream>

#include<iostream>

using namespace std;

/\* Function to sort an array

using insertion sort\*/

void insertionSort(int arr[], int n)

{

int i, key, j;

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

{

key = arr[i];

j = i - 1;

/\* Move elements of arr[0..i-1],

that are greater than key,

to one position ahead of

their current position \*/

while (j >= 0 && arr[j] > key)

{

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

// A utility function to print

// an array of size n

void printArray(int arr[], int n)

{

int i;

for (i = 0; i < n; i++)

printf("%d ", arr[i]);

printf("\n");

}

// Driver code

int main()

{

ofstream MyFile("executionTime.csv");

for(int n = 1; n <= 100000; n\*=2)

{

int arr[n];

for(int i = 0; i < n; i++)

{

arr[i] = rand();

}

clock\_t tStart = clock();

insertionSort(arr, n);

double time1=(double)(clock() - tStart)/CLOCKS\_PER\_SEC;

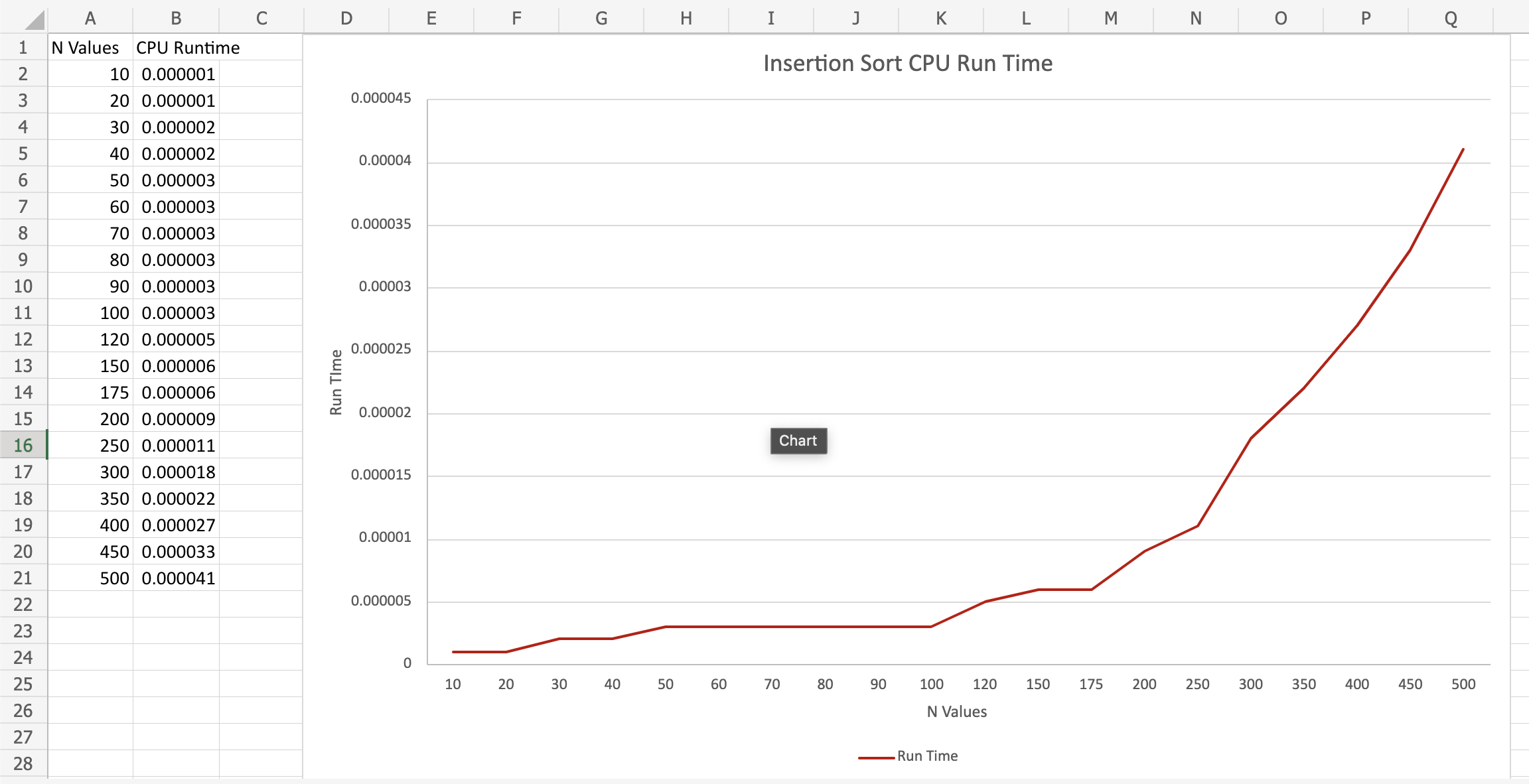
MyFile << n << "," << time1<< endl;

}

return 0;

}

Output:



**Program 1. Define function to find first digit of a number**

Code:

#include<iostream>

using namespace std;

void PrintLastDigit(int n){

if(n > 10){

PrintLastDigit(n/10);

}

else{

cout<<n;

}

}

int main()

{

int n;

std::cout<<"Enter the number : ";

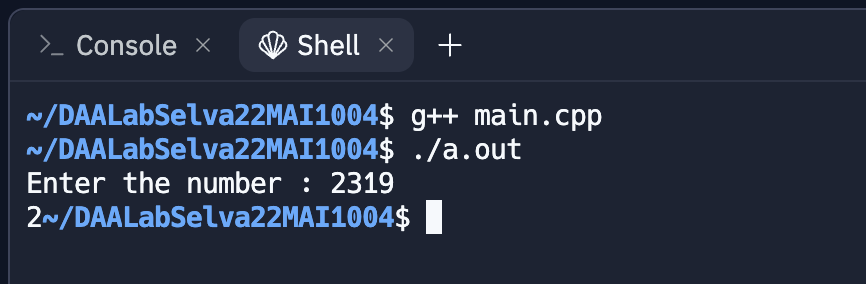
std::cin>>n;

PrintLastDigit(n);

return 0;

}

Output:



**Program 2. Define function to find second digit of a number.**

Code:

#include<iostream>

using namespace std;

void PrintSecondDigit(int n)

{

if(n > 100)

{

PrintSecondDigit(n/10);

}

else

cout<<n%10;

}

int main()

{

int n;

cout<<"Enter the Number : ";

cin>>n;

if(n < 10){

cout<<"Please enter numbers having atleast 2 digits!! Try again.."<<endl;

}

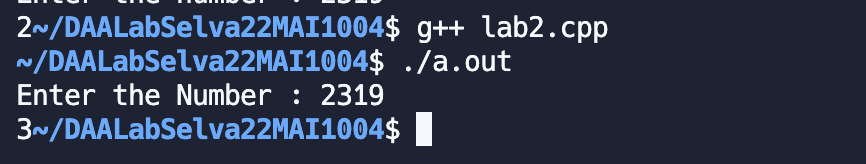
else

PrintSecondDigit(n);

return 0;

}

Output:



**Program 3. Define function to find the number of digits.**

Code:

#include<iostream>

using namespace std;

int GetNumberOfDigits(int n){

if(n < 10){

return 1;

}

return 1 + GetNumberOfDigits(n / 10);

}

int main(){

int n;

cout<<"Enter the number : ";

cin>>n;

cout<<GetNumberOfDigits(n);

return 0;

}

Output:

Graphical user interface, text, application

Description automatically generated

**Program 4. Define function to find sum of digits except second digit. sum(231456)=2+1+4+5+6=18.**

Code:

#include<iostream>

using namespace std;

int SumDigits(int n){

if(n < 10)

return n;

if(n > 10 && n < 100){

return SumDigits(n/10);

}

return (n % 10) + SumDigits(n / 10);

}

int main(){

int n;

cout<<"Enter the num ";

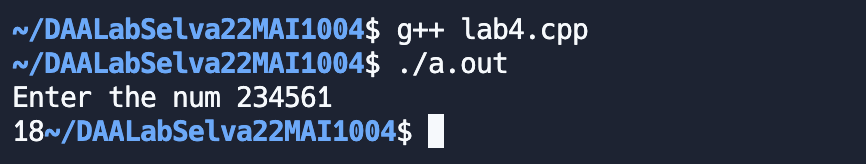
cin>>n;

cout<<SumDigits(n);

return 0;

}

Output:



**Program 5. Define function to find last even digit. f(354683257) returns 2**

Code:

#include<iostream>

using namespace std;

void PrintLastEven(int n){

if((n % 10) % 2 == 0){

cout<< n % 10;

}

else{

PrintLastEven(n / 10);

}

}

int main(){

int n;

cout<<"Enter the Number : ";

cin>>n;

PrintLastEven(n);

return 0;

}

Output:

Graphical user interface, text, application

Description automatically generated with medium confidence

**Program 6. Define function to find digit before last even digit. f(354683257) returns 3**

Code:

#include<iostream>

using namespace std;

void PrintLastEven(int n){

if((n % 10) % 2 == 0){

if(n > 10)

cout<< (n / 10) % 10;

}

else{

PrintLastEven(n / 10);

}

}

int main(){

int n;

cout<<"Enter the Number : ";

cin>>n;

PrintLastEven(n);

return 0;

}

Output:

Graphical user interface, text, application

Description automatically generated

**Program 7. Define function to find digit after last even digit. f(354683257) returns 5.**

Code:

#include <iostream>

using namespace std;

void PrintLastEven(int n){

if(n > 10 && ((n / 10) % 10) % 2 == 0){

cout<< n % 10;

}

else{

PrintLastEven(n / 10);

}

}

int main(){

int n;

cout<<"Enter the Number : ";

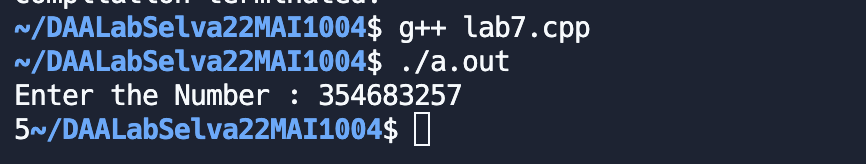
cin>>n;

PrintLastEven(n);

return 0;

}

Output:



**Program 8. Define function to find the location of the last even digit from last. e.g. in 354683257 the last even digit is 2 and its location from last is 3.**

Code:

#include<iostream>

using namespace std;

void PrintLastEven(int n, int pos){

if((n % 10) % 2 == 0){

cout<< "Digit : "<< n % 10 << " Location : " << pos;

}

else{

PrintLastEven(n / 10, pos+1);

}

}

int main(){

int n;

cout<<"Enter the Number : ";

cin>>n;

PrintLastEven(n, 1);

return 0;

}

Output:

Text

Description automatically generated

**Program 9. Define function int g(int x, int k). It finds kth digit (from last) of number x. g(2536487,3) returns 4.**

Code:

#include<iostream>

using namespace std;

void PrintLastEven(int n, int pos, int currentPos){

if(pos == currentPos){

cout<< "Digit : "<< n % 10;

}

else{

PrintLastEven(n / 10, pos, currentPos + 1);

}

}

int main(){

int n;

int pos;

cout<<"Enter the Number : ";

cin>>n;

cout<<"Enter the position : ";

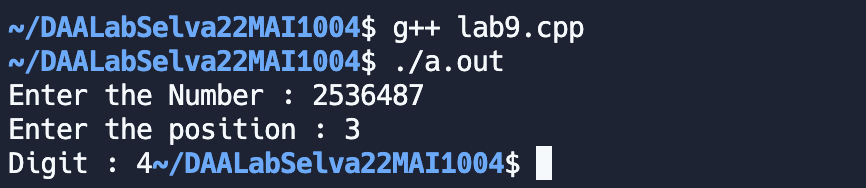
cin>>pos;

PrintLastEven(n, pos, 1);

return 0;

}

Output:



**Program 10. Define function to find sum of even digits.**

Code:

#include<iostream>

using namespace std;

int SumEvenDigits(int n){

if(n < 10){

if(n % 2 == 0) return n;

return 0;

}

if((n % 10) % 2 == 0){

return (n % 10) + SumEvenDigits(n / 10);

}

return SumEvenDigits(n / 10);

}

int main(){

int n;

cout<<"Enter the Number : ";

cin>>n;

cout<<SumEvenDigits(n);

return 0;

}

Output:

Graphical user interface, application

Description automatically generated