**Program 1: Addition of two Arrays:**

**Serial Execution**

Program

**--------------------------------------------------------------------------------------------**

**#include<iostream>**

**#include <stdlib.h>**

**#include<time.h>**

**#define size 100000**

**using namespace std;**

**int main()**

**{**

**clock\_t t;**

**t = clock();**

**int arr1[size];**

**int arr2[size];**

**int sum[size];**

**for(int i = 0; i < size; i++) //assigning for first array.**

**{**

**arr1[i] = rand()%size + 1;**

**}**

**for(int i = 0; i < size; i++) //assigning for second array.**

**{**

**arr2[i] = rand()%size + 1;**

**}**

**for(int i = 0; i < size; i++)**

**{**

**sum[i] = arr1[i] + arr2[i];**

**}**

**double time\_taken = ((double)t)/CLOCKS\_PER\_SEC;**

**cout<<"\nTook "<<time\_taken<<" seconds to execute \n";**

**return 0**

**}**

**Output:**

**--------------------------------------------------------------------------------**

**Took 172.656 seconds to execute**

**--------------------------------------------------------------------------------**

**Parallel execution**

Program

**--------------------------------------------------------------------------------------------**

**#include<iostream>**

**#include <stdlib.h>**

**#include<time.h>**

**#include<omp.h>**

**#define size 100000**

**using namespace std;**

**int main()**

**{**

**clock\_t t;**

**t = clock();**

**int arr1[size];**

**int arr2[size];**

**int sum[size];**

**#pragma omp parallel for**

**for(int i = 0; i < size; i++) //assigning for first array.**

**{**

**arr1[i] = rand()%size + 1;**

**}**

**#pragma omp parallel for**

**for(int i = 0; i < size; i++) //assigning for second array.**

**{**

**arr2[i] = rand()%size + 1;**

**}**

**#pragma omp parallel for**

**for(int i = 0; i < size; i++)**

**{**

**sum[i] = arr1[i] + arr2[i];**

**}**

**double time\_taken = ((double)t)/CLOCKS\_PER\_SEC;**

**cout<<"\nTook "<<time\_taken<<" seconds to execute \n";**

**return 0; }**

**--------------------------------------------------------------------------------**

**Output:**

**--------------------------------------------------------------------------------**

**Took 164.918 seconds to execute**